UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

DEVELOPMENT AND ANALYSIS OF ALGORITHMIC IMPROVEMENTS FOR THE ADCIRC HYDRODYNAMIC MODEL

A Dissertation

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

degree of

Doctor of Philosophy

By

CHRISTINE MICHELLE SZPILKA
Norman, Oklahoma
2005

UMI Number: 3174277

UMI Microform 3174277

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

DEVELOPMENT AND ANALYSIS OF ALGORITHMIC IMPROVEMENTS FOR THE ADCIRC HYDRODYNAMIC MODEL

A Dissertation APPROVED FOR THE
SCHOOL OF CIVIL ENGINEERING AND ENVIRONMENTAL SCIENCE

By

_____

Randall L. Kolar

_____

Clint N. Dawson

_____

Kyran D. Mish

_____

Baxter E. Vieux

_____

Luther W. White

Acknowledgements

I could not have completed this work without the help and support of a great many people. This dissertation is dedicated to those who have supported me throughout this endeavor:

To my husband, for your encouragement and tenacity in pushing me to finish, all the while putting up with my complaints and stress: I love you and am grateful for all of your support! To my son for providing many breaks from the work and always giving me something to laugh about: thanks for loving me unconditionally! To my parents for providing moral support, baby-sitting services, and a lifetime of love: I am who I am largely due to your guidance throughout the years (so don't blame me!). To my brother and sister, grandparents, and extended family for their encouragement and love: thanks for always being there when I needed to talk.

To the people in the background who just made life a little more interesting and fun. To the rest of the "back corner group" (Kendra, Melissa and Casey) for always being there to provide a much needed distraction and more than a few laughs: even though our lives now go in different directions, I will always treasure your friendships. To the "office girls" (Susan, Molly, Brenda and Audre) for helping me navigate the university red tape and always being ready with encouragement and laughs: thanks for everything! To the many nameless, including the parish of Saint Joseph, who befriended a misplaced Montana girl and made life in Oklahoma an enjoyable experience (when possible). And to Paula Hopwood, for loving me like a daughter and inspiring me through your selfless attitude: you will be sorely missed and lovingly remembered.

To my committee members, the people who advised and guided me from the proposal to the defense: thanks for reading my 500 page epistle and mostly for passing me!

Finally, to my advisor for not only guiding me through seven long years of classes and research, but mostly for being a great friend, adopting me into your family, sponsoring me at St. Joe's, and teaching me the importance of balance in life: thanks for everything and then some!

This material is based upon work supported under a National Science Foundation Graduate Fellowship. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the author and do not necessarily reflect the views of the National Science Foundation. Additional funding was also provided from an OU Alumni Fellowship and under National Science Foundation Grant ACI-9623592.

Table of Contents

List of Tables

List of Figures

Abstract

This work addresses enhancements to the popular shallow water model, ADCIRC, ADvanced three-dimensional CIRCulation model. The shallow water equations (SWE) are based on conservation of mass and momentum and can be used to model the hydrodynamic behavior of oceans, coastal areas, estuaries and lakes. In this dissertation, four research areas have been identified in an effort to improve the predictive capabilities of ADCIRC.

1) Propagation analysis is a useful tool to examine the simulation characteristics of an algorithm. Two analytical tools exist for this analysis: Fourier and dispersion analysis. Unfortunately, it is not always possible to write an algorithm in the required closed form for these analyses. Therefore, a numerical analog of these tools was developed for this study. Three initialization techniques and three wave tracking schemes were examined. Results show that no single combination of initialization and tracking schemes is able to predict both the Fourier and dispersion characteristics of an algorithm. However, by examining several combinations the full behavior can be predicted. Additionally, these tools now provide a structured mechanism for evaluating slope limiters within the finite volume and discontinuous Galerkin frameworks.

2) Currently, the ADCIRC model suffers from local mass conservation errors is regions of high advection. Meanwhile, the discontinuous Galerkin (DG) finite element method is relatively new to the SWE modeling effort and has several appealing properties including: local mass conservation, shock capturing without spurious oscillations, and local *h*- and *p*- refinement. The numerical tools that were developed in topic 1 were used to compare the propagation behavior of the DG algorithm with the ADCIRC model. Results indicate that the DG algorithm has similar propagation behavior to the ADCIRC model. Namely, very little dissipation of the physical waves and no spurious oscillations (when

higher-order interpolants are used). Additionally, the higher-order approximations have near second-order spatial convergence rates and smaller errors than the ADCIRC model.

3) However, the higher-order DG algorithm is also computationally costly due to the extra degrees of freedom. Therefore, multi-algorithmic models, which couple continuous and discontinuous algorithms in the solution of the SWE, were examined in the hope of balancing accuracy with computational costs. Three levels of coupling were examined: equation, subdomain, and combined coupling. Propagation results indicate that the low-order versions are over-dissipative in the physical wavelengths. Furthermore, the equation and combined coupled models allow trailing wiggles if the higher-order approximations are not limited. Convergence results indicate that the subdomain-coupling scheme has similar accuracy when compared to the domain-wide DG model; however, results from an application on the East coast indicate that it does not reduce the CPU usage for 1D problems. Further work using 2D applications is needed to determine what the CPU affects of the coupled model would be in practice. Additionally, the coupling mechanism itself needs to be examined further, and flux coupling should be examined as an alternative to overlapping elements.

4) Baroclinic models that are used to simulate density driven flows require a mass conservative transport module in order to determine the new density profile, which is due to the changing velocity field. In this study, the local DG (LDG) method is applied to the advective-diffusive transport equation and added to the ADCIRC model as a module. The LDG transport model was verified with an analytical solution for breakthrough curves, and was found to be stable and mass conservative from advection-dominated flows to diffusion-dominated flows. Furthermore, when added as a module to create a baroclinic ADCIRC model, it remains stable and the DG salinity fields are mass conservative. However, the interpolation of the DG salinities to the CG gridspace is not mass conservative; and the communication between the continuous ADCIRC model and the discontinuous transport module should be examined further.

CHAPTER 1

Study background and motivation

1.1 Introduction

The overall goal of this study is to improve the state of shallow water modeling in two broad aspects. The first is to improve the accuracy of the production code, ADCIRC (ADvanced 3D CIRCulation model *Luettich et al.* [1992, 2003]), by studying continuous and discontinuous discretization algorithms and various coupling mechanisms of the two. The hope is to improve the overall quality of ADCIRC solutions by using multiple algorithms within their appropriate subdomains in order to maintain computational efficiency while increasing accuracy. The second is to increase the range of physical applications that ADCIRC can accurately model by coupling a mass conservative transport algorithm to the existing hydrodynamics. This will allow the model to handle prognostic baroclinic simulations, where the salinity and temperature are allowed to advect due to the hydrodynamics and the pressure gradients for the hydrodynamics are updated accordingly.

1.2 Shallow water modeling

The shallow water equations (SWE) are derived from the depth-averaged Navier-Stokes equations under the Boussinesq approximation (assumes that the density does not depart much from a mean value) and also assuming hydrostatic pressure and a vertically uniform horizontal velocity profile. The SWE describe the motion of water bodies wherein

the depth is small relative to the scale of the waves propagating on that body. The system of SWE consists of a continuity equation with either a conservative or non-conservative form of the momentum equation and is used to model the hydrodynamics of lakes, estuaries, coastal regions, and other impoundments, as well as deep ocean tides. Numerical techniques are required to solve the SWE over complex domains, and it is desirable that these algorithms produce accurate results without introducing numerical noise (spurious oscillations) into the solution.

The wide range of applications used in shallow water modeling necessitate robust and computationally-efficient algorithms that maintain a high degree of accuracy. For example, storm surge predictions require immediate results with a certain degree of accuracy, while coastal dredging applications require accuracy but not immediate results. All applications might not be adequately served with a single algorithm, therefore a detailed understanding of each algorithm's shortcomings and advantages is important for determining which should be used for a given application. Meanwhile, numerical models that incorporate more of the physical processes will increase the valid range of applications.

1.2.1 Shallow water equations in 2D

The primitive 2D (two-dimensional) system of depth-averaged SWE consists of a continuity equation and a momentum equation. In operator form, the primitive form of the continuity equation, L, is

$$L \equiv \frac{\partial \zeta}{\partial t} + \nabla \bullet (H\mathbf{V}) = 0 \qquad \textbf{(1.1)}$$

where ζ is surface elevation above a datum, H is the total fluid depth, and $\mathbf{V}$ is the 2D depth-averaged velocity. The conservative form of momentum conservation, $\mathbf{M}^{\mathrm{c}}$, is given by

$$\mathbf{M}^{\mathrm{c}} \equiv \frac{\partial (H\mathbf{V})}{\partial t} + \nabla \bullet (H\mathbf{V}\mathbf{V}) + \tau H\mathbf{V} + H\mathbf{f} \times \mathbf{V} + H\nabla\left[\frac{p_{\mathrm{a}}}{\rho} + g(\zeta - \alpha\eta)\right] -$$

$$\mathbf{K} - \mathbf{D} - \mathbf{B} - \frac{1}{\rho}(\mathbf{T}_b - \mathbf{T}_s) = 0 \qquad \textbf{(1.2)}$$

where τ is the bottom friction parameter, **f** is the Coriolis parameter, p_a is the atmospheric pressure at the surface, α is the effective earth elasticity factor, η is the Newtonian equilibrium tidal potential, ρ is the density of water, g is gravity, **K** is the depth-integrated horizontal momentum diffusion, **D** is the momentum dispersion, **B** is the depth-integrated baroclinic forcing, $\mathbf{T}_b$ is the bottom stress, and $\mathbf{T}_s$ is the surface stress. The non-conservative form of the momentum equation is obtained by subtracting $\mathbf{V}L$ from (1.2) and dividing the result by H to get

$$\mathbf{M} \equiv \frac{\partial \mathbf{V}}{\partial t} + \mathbf{V}\bullet\nabla\mathbf{V} + \tau\mathbf{V} + \mathbf{f}\times\mathbf{V} + \nabla\left[\frac{p_a}{\rho} + g(\zeta - \alpha\eta)\right] - \frac{1}{H}(\mathbf{K} + \mathbf{D} + \mathbf{B}) -$$

$$\frac{1}{\rho H}(\mathbf{T}_b - \mathbf{T}_s) = 0 \qquad \textbf{(1.3)}$$

The generalized wave continuity (GWC) equation, is formed from the following operation:

$$\frac{\partial L}{\partial t} + GL - \nabla\bullet\mathbf{M}^c = 0 \qquad \textbf{(1.4)}$$

where G is a numerical parameter. In the ADCIRC model, it is used in lieu of (1.1) to solve for free surface elevation. Note that the wave continuity equation, as it originally appeared in *Lynch and Gray* [1979], is obtained by setting $G = \tau$.

1.2.2 Linearized shallow water equations in 1D

For analysis purposes, the linearized 1D (one-dimensional) form of these equations are used for most of this study. Note that in one spatial dimension, the conservative and non-conservative momentum equations are equivalent after linearization. Therefore, in the remainder of this dissertation, conservation of momentum will be referred to simply as the momentum equation, instead of qualifying it with conservative or non-conservative, when

the 1D equations are referenced.

The system of SWE in 1D consists of the primitive continuity equation

$$L \equiv \zeta_t + hu_x = 0 \qquad \textbf{(1.5)}$$

and the momentum equation

$$\mathrm{M} \equiv u_t + \tau u + g\zeta_x = 0\,, \qquad \textbf{(1.6)}$$

where u is the depth-averaged velocity, ζ is the surface elevation, τ is the linear bottom friction factor, g is the acceleration of gravity, h is the bathymetric depth of water (assumed constant), and the subscripts indicate partial derivatives. The linearized form of the generalized wave continuity (GWC) equation is given as

$$\zeta_{tt} + G\zeta_t + (G-\tau)hu_x - gh\zeta_{xx} = 0 \qquad \textbf{(1.7)}$$

where G is a numerical coefficient that determines the balance between primitive (large values of G) and pure wave (small values of G) forms.

1.2.3 History of shallow water solution algorithms

One of the first successful discretizations of the primitive equations that did not produce spurious modes was the staggered finite difference (SFD) scheme of *Leendertse* [1967]. This approach uses a staggered central difference stencil on a regular grid with the velocity evaluated at the grid interfaces and elevation centered in the middle of the grid block. This ensures perfect mass balance at local and global levels.

Early finite element algorithms used to discretize the primitive system of 2D SWE (depth-averaged, unless indicated otherwise) often resulted in either spurious oscillations or artificial damping of the solution. Simple linear Galerkin finite element discretizations, henceforth referred to as the primitive finite element (PFE) method, were plagued by spurious oscillations. Various methods were introduced to eliminate these oscillations, but

these often incorporated artificial or numerical damping, thus damping out the physical solution in addition to the unwanted noise. Mixed finite elements were used by *Hood and Taylor* [1974] in an attempt to mimic the success of the SFD scheme, but *Walters* [1983] indicates that noise still remains in the velocity solution.

Motivated by the generally poor results from the primitive forms of the SWE, another technique was developed by *Lynch and Gray* [1979], which modified the system of SWE before trying numerical solutions. It was proposed to use the newly-derived wave continuity equation instead of the primitive continuity in the shallow water model. This approach was further modified with the addition of a numerical parameter, *G*, by *Kinnmark* [1986] in order to obtain the generalized wave continuity (GWC) equation, which is the form studied in this dissertation. The resulting GWC equation and the non-conservative momentum equation are then discretized using piecewise linear Galerkin finite elements in space. These wave continuity methods have proven to eliminate the presence of spurious oscillations, as verified by *Foreman* [1983] and *Kolar et al.* [1994]. It is also notable that in the limit, as *G* approaches infinity, the GWC equation reduces to the primitive continuity form.

Another finite element technique of the same generation is the selective lumping finite element (SLFE) scheme of *Kawahara et al.* [1982]. This is a two-step explicit scheme in which the mass matrix is selectively lumped by using a lumping parameter with standard linear Galerkin elements. The parameter can take on values between 0.0 (consistent mass matrix) and 1.0 (fully lumped mass matrix), where in-between values "selectively" lump the mass matrix. Kawahara found that the algorithm was free from artificial numerical damping when the maximum stable time step was used and the lumping parameter took on values between 0.8 and 0.95. In this current study, the procedure has been further generalized by adding linear bottom friction to Kawahara's formulation and introducing an additional lumping parameter for the mass matrix associated with this term, thus allowing greater flexibility in the analyses.

More recently, *Alcrudo and Garcia-Navarro* [1993] and *Chippada et al.* [1998] have applied a Godunov-type finite volume method (FVM) to the system of SWE. The FVM has enjoyed use in gas dynamics related to mechanical and aerospace engineering applications, but it has not seen widespread use in hydrodynamic modeling. The method applies local conservation of mass and momentum by integrating over a discrete volume, rather than the entire domain, and allows the state values to be discontinuous at volume interfaces. Due to these discontinuities at the interfaces, the boundary fluxes on each volume are not uniquely defined, and the Riemann problem must be solved at each interface. The combined effects of round-off error and computational inefficiency leave little reason to solve the Riemann problem exactly, and therefore an approximate solver is often used in practice. The cell-centered solution that results from this method consists of the average state values located at the center of each discrete volume. Higher-order results can be obtained by post-processing these average values through slope reconstruction and limiting steps.

Advantages of the FVM technique include its ability to capture shocks without introducing spurious oscillations, local and global mass conservation, utilization of the primitive form of the equations, and the ability to handle irregular meshes. The FVM low-order schemes use piecewise constant approximations within each volume, and higher-order schemes use piecewise linear (or higher) approximations. However, unlike finite element methods, which use polynomial interpolants for the state variables, the higher orders in FVM approximations are added through post-processing reconstruction of the volume averages. Additionally, the high-order schemes require a limiting procedure to control the non-physical oscillations that may be introduced in the solution; the nature of these limiting schemes (i.e., evolution with the solution at each time step) does not permit the higher-order FVM to be analyzed using traditional analytical techniques, so numerical approaches must be used instead.

There are several approximate Riemann solvers available that can be implemented

in the FVM solution scheme. However, the most popular is attributed to *Roe* [1981]. The basic idea is to solve a system of linear, constant coefficient conservation laws instead of the original non-linear system. The coefficient matrix is dependent upon the left and right states at the interface, and the corresponding approximate Riemann solution is a linear combination of this matrix's eigenvectors.

Additionally, there are various limiting schemes available for piecewise linear high-order FVM approximations. *Sweby* [1984] has examined Roe's minmod and superbee limiters, as well as vanLeer's limiter, in the context of flux limiting and found that these limiters all satisfy the total variation diminishing inequalities, which is a sufficient condition to ensure that an algorithm is monotonicity preserving (i.e., does not introduce oscillations into the solution). *Leveque* [1990] also discusses the minmod and van Leer limiters. *Causon et al.* [2000] have used the superbee and van Leer limiter in their cut cell finite volume solution to the shallow water equations, and *Bell and Shubin* [1984] have used the van Leer limiter in reservoir simulation applications. In response to solution difficulties, other specialized limiting techniques have been developed in the context of finite volume solutions by *Chippada et al.* [1998].

Another solution technique that has recently been applied to the system of SWE by *Aizinger and Dawson* [2002] and to the compressible Navier-Stokes equations by *Bassi and Rebay* [1997], as well as others, is discontinuous Galerkin (DG) finite element spaces. These algorithms are derived in a similar manner to the finite volume scheme in that the primitive equations are integrated over individual elements, rather than the entire global domain, and the state values are allowed to be discontinuous at element interfaces. In fact, one can think of the DG scheme as being a generalization of earlier Godunov finite volume algorithms. However, DG methods are more similar to traditional finite element techniques in that the form of the unknown states are specified beforehand by piecewise discontinuous basis functions within a weighted residual statement, thus eliminating the need for ad-hoc post-processing, as in FVM schemes. This framework also allows different orders of

approximating spaces to be used in adjacent elements since each element is integrated independently of its neighbors; thus the algorithm can easily adapt and incorporate higher resolution (i.e., *p*-refinement) near shocks in the solution. Other useful features of DG methods include the ability to handle second-order diffusive terms, which is not possible in the Godunov FVM framework, and the ability to handle non-conforming grids (i.e., grids with non-matching faces). As in Godunov-type finite volume schemes, DG methods require solution of the Riemann problem at the element interfaces and approximate solvers are employed in practice.

For first-order equations, all DG methods are equivalent, however several discontinuous techniques have been developed for handling second-order equations. One of the more popular methods is the local discontinuous Galerkin (LDG) method, as developed and analyzed by *Cockburn and Shu* [1998b]. This method is an extension of the Runge-Kutta DG method for conservation laws that was presented and analyzed in a series of papers (*Cockburn and Shu* [1989, 1991, 1998a], *Cockburn et al.* [1989, 1990]). This is the form of the method used by *Aizinger and Dawson* [2002] in their application to the system of SWE. *Bassi and Rebay* [1997] use a hybrid combination of the LDG and a mixed finite element formulation similar to the *Raviart-Thomas* [1986] formulation. *Cockburn et al.* [2002] provide an excellent history of the development of DG methods.

1.3 Current deficiencies in SWE modeling

In this study, two general deficiencies in SWE modeling are examined: simulation errors, instabilities and efficiency; and over-simplification of physical processes. The first is an artifact of the numerical algorithm itself, while the second lies in the development of the model. One specific problem that frequently occurs when using continuous finite element algorithms to model regions of high advection is mass balance errors. This error presents itself as a non-physical source or sink of water and affects the validity of the model results for elevation and velocity profiles, as well as transport processes. Likewise, model

instabilities can present themselves as unphysical oscillations in the simulation results and often lead to the eventual failure of the run.

Additionally, the simplifications made during model development often lead to a solution that cannot adequately match physical processes. One such area of shallow water modeling that is in need of further investigation is prognostic baroclinic simulations. The ability to incorporate the effects of salinity and temperature transport in the ocean, and thus density differences, would provide a more complete physical representation of the actual processes that affect circulation. This requires an accurate representation of the salinity and temperature transport throughout the domain, which is affected both by the quality of the transport algorithm itself and the accuracy of the hydrodynamic results (velocities) that drive the transport. Thus, the transport model must be dynamically coupled with the hydrodynamic model.

Since many model deficiencies are a function of the algorithm, a comparative study of the mathematical properties of various algorithms is conducted in this study. Several analytical analysis tools exist for examining the numerical properties of linear algorithms. Of the available techniques, Fourier, dispersion, and truncation error analyses are used to examine the propagation behavior, stability and accuracy of linear algorithms. In Chapter 2, numerical counterparts to these tools are developed so that the behavior of algorithms that evolve with the solution (such as higher-order DG), and thus cannot be written in closed form, can be examined. In Chapter 3, a combination of analytical and numerical tools are used to compare several SWE algorithms, as applied to the linear 1D SWE, in order to determine the strengths and weaknesses of each. The algorithms of interest for this study are as follows: staggered finite difference, primitive finite element, generalized wave continuity, selective lumping finite element, finite volume, and discontinuous Galerkin finite element. In Chapter 4, the properties of subdomain and equation multi-algorithmic models for the SWE are examined. In subdomain multi-algorithmic models, different algorithms are employed in unique partitions, or subdomains, of the larger domain in order

to exploit each algorithm's best features, while minimizing their weaknesses. For equation multi-algorithmic models, different algorithms are used to discretize the continuity and momentum equations within the same domain.

Finally, as a first step toward a full three-dimensional baroclinic model, a quasi-3D (*x-z* slice) DG transport algorithm is developed and incorporated into an ADCIRC hydrodynamic model using the same coordinates in Chapter 5. Various test cases are explored in this laterally-averaged framework to allow for faster testing and comparisons between different algorithms before proceeding to a full 3D framework.

Each of the remaining chapters in this dissertation present the background and findings of four individual projects, which culminated in individual papers (proceedings and/or journal or to be submitted). Therefore, there is some redundancy in the literature review and background sections within these chapters. However, since many of the chapters reference the same material, all of the references are collected together at the end of the dissertation rather than repeating them for each chapter.

1.4 Dissertation "roadmap"

As stated above, this study seeks to address two main issues in the existing ADCIRC hydrodynamic model: 1) local mass conservation and stability issues in highly advective regions, and 2) the need for a mass conservative transport module for full baroclinic (density driven) flows. For the first issue, two possible solutions are examined, while one possible solution is examined for the last issue. Thus, this dissertation can be broken up according to the "big picture" as follows:

Issue 1 *Local mass conservation and stability issues in highly advective regions*

Possible solution 1: use another algorithm altogether?

Chapter 2 – develop numerical analysis tools to use in one-dimensional algorithm comparisons

Chapter 3 – analyze and compare several shallow water solution algorithms in one dimension

Possible solution 2: use coupled discontinuous and continuous models?

Chapter 4 – develop and test three coupled models in one spatial dimension in order to balance computational costs and accuracy by selectively using discontinuous algorithms

Issue 2 *Mass conservative transport module for full baroclinic flows*

Possible solution: add LDG transport module?

Chapter 5 – develop and test local discontinuous Galerkin transport algorithm in laterally averaged x-z and x-σ grid coordinates

CHAPTER 2

Numerical Analogs to Fourier and Dispersion Analysis: Development, Verification, and Application to the Shallow Water Equations

2.1 Introduction

The shallow water equations (SWE), as introduced in Chapter 1, have been successfully discretized using the staggered finite difference (SFD) method developed by *Leendertse* [1967], the selective lumping finite element (SLFE) method of *Kawahara et al.* [1982], and the finite volume method (FVM) (*Alcrudo and Garcia-Navarro* [1993], *Causon et al.* [2000], *Chippada et al.* [1998]). More recently, discontinuous Galerkin (DG) finite element methods, which can be thought of as a generalization of the FV methods, have been used by *Aizinger and Dawson* [2002] and *Bassi and Rebay* [1997]. In addition, Galerkin finite element approximations of modified equations, such as the wave continuity approach developed by *Lynch and Gray* [1979] and later generalized by *Kinnmark* [1986], have been used. Primitive finite element (FE) discretizations of the SWE were not as successful and often introduced numerical noise into the solution. Accurate solutions demand that the algorithm not introduce spurious modes or overdamp the system. In an effort to characterize these features of a successful algorithm, much work has been done to study the propagation characteristics of the FE and FD discretizations in both 1D and 2D frameworks (*Atkinson et al.* [2004], *Foreman* [1983, 1984], *Gray and Lynch* [1977],

Kinnmark and Gray [1985], *Kolar et al.* [1994], *Leendertse* [1967], *Lynch and Gray* [1979], *Platzman* [1981], *Walters* [1983], *Westerink et al.* [1994]); but similar studies cannot be found for the FV, DG and SLFE methods, as applied to the SWE. (*Hu et al.* [1999] examine the dispersion properties of the DG method applied to the 1D advection equation and 2D wave equation.)

Analytical Fourier and dispersion analyses can be used to study the propagation characteristics of many discretizations of the SWE. Such analyses provide information about the phasing of the numerical solutions, the damping characteristics and the stability of the discretization method. This information is valuable for determining whether an algorithm is suitable for various applications. As model applications become more complex and simulation domains more refined along the coastlines, many current discretization algorithms experience instabilities. The FVM and DG class of discretizations are being examined to determine if they are a suitable method for near-shore applications, where the more shallow water produces greater instabilities due to the increased importance of the nonlinear terms in the equations of motion and due to the possibility of shocks forming. However, some discretization methods are not suitable for closed form analyses. For example, higher-order FVM discretizations, which use piecewise linear or higher interpolates, require some type of slope limiting procedure. These algorithms are transient, due to the slope limiting, and cannot be written in a closed form for all time. Consequently, numerical analogs to the traditional analysis techniques are developed herein.

With the exception of *Kinnmark and Gray's* $2\Delta x$-test [1985], which only focused on a single wavelength, most of the previous propagation work has focused on the use of analytical analysis methods. In this study, numerical analogs for generating Fourier and dispersion plots are developed. Several benefits exist for undertaking a numerical approach: 1) it is possible to study boundary effects, which are often ignored in analytical methods; 2) if the algorithm evolves with time, it is not possible to write the closed form equation required for analytical analysis; and 3) numerical phase results are more closely

tied to the algorithm output and thus capture more of the actual simulation behavior relative to the predicted mathematical behavior (including any unforeseen effects of truncation and roundoff error).

In the next section, the traditional analytical propagation analysis tools are presented. The procedure for the numerical analogs to these traditional tools is presented in §2.3. Finally, the validation results are presented in §2.4 and some closing remarks are offered in §2.5.

2.2 Analytical Propagation Analysis

Two frequently used tools for analytical propagation analysis exist: Fourier analysis, which uses the fully discretized equations; and dispersion analysis, which discretizes only the spatial variables, leaving time continuous via the harmonic form of the equations. As will be seen in §2.3.2, these two can be related through the phase speed. The desired propagation characteristics for any algorithm can be found by looking at the continuum equations, which, in this case, exhibit a monotonic dispersion relationship. It is also desirable for the algorithm to damp only those waves that propagate with significant phase error (typically high-frequency components on the order of 2-5Δx).

2.2.1 Fourier analysis

Fourier analysis is a useful tool to analyze the propagation behavior of both analytical and discrete solutions of homogeneous, linear difference or differential equations. It utilizes the assumption that separation of variables can be used to express the solution as periodic in time and space, and thus any solution that is periodic on an infinite interval or piecewise continuous on a finite interval can be represented by a Fourier series expansion. A solution (discrete or analytical) that satisfies these minimal constraints can be represented as a complex Fourier series

$$b(x,t) = \sum_{n=-\infty}^{\infty} A_n e^{i\omega_n t} e^{i\sigma_n x} \qquad \textbf{(2.1)}$$

where, for the SWE, $b = u$ or ζ, ω_n is the temporal frequency of the solution, $\sigma_n = 2\pi/L_n$ is the wave number and L_n is the n^{th} wavelength, A_n is the Fourier coefficient for component n and $i^2 = -1$. The key to Fourier analysis is that the linearity of the difference or differential equation allows one to examine a single Fourier component of the solution at a time. Such an analysis, also known as von Neumann analysis, results in two related plots: phase error versus wavelength and damping ratio versus wavelength. Similarly, one can introduce a complex propagation factor, T, which is the ratio of the discrete wave to the continuum wave after the time it takes for the continuum wave to propagate one wavelength, as was done by *Gray and Lynch* [1977] and *Leendertse* [1967]. The latter is the approach that will be taken in this study.

The desired behavior of a solution algorithm is that it propagates all waves in-phase, but for those waves that are out of phase, sufficient damping should be provided; too much damping results in an overly-dissipative system, while too little damping (combined with out-of-phase propagation) results in the introduction of numerical noise in the solution. Additionally, a damping ratio of magnitude greater than one indicates an unstable solution, since any noise will be amplified at each time step until the solution eventually becomes infinite.

The starting point for such an analysis is to determine the wave properties from the continuum equations. Recognizing that the continuum equations are linear, one can drop the subscripts n in Equation (2.1) and substitute a single component into the continuum equations, (1.5) and (1.6), which results in the following system of equations:

$$i\omega\zeta_0 e^{i(\omega t + \sigma x)} + i\sigma h u_0 e^{i(\omega t + \sigma x)} = 0 \qquad \textbf{(2.2)}$$

$$i\omega u_0 e^{i(\omega t + \sigma x)} + \tau u_0 e^{i(\omega t + \sigma x)} + i\sigma g\zeta_0 e^{i(\omega t + \sigma x)} = 0. \quad \textbf{(2.3)}$$

These can be further simplified by dividing out the exponential factor to get

$$i\omega\zeta_0 + i\sigma h u_0 = 0 \text{ and} \quad \textbf{(2.4)}$$

$$i\omega u_0 + \tau u_0 + i\sigma g\zeta_0 = 0. \quad \textbf{(2.5)}$$

The eigenvalues of this system can be used to eliminate one equation and solve for the elevation and velocity as a function of time, thus giving an analytical solution to the continuum equations. These eigenvalues can be represented as

$$\omega = \frac{1}{2}[i\tau \pm \sqrt{4\sigma^2 gh - \tau^2}] \quad \textbf{(2.6)}$$

which allows the solution to be written as

$$\zeta(x, t) = \zeta^o e^{-\tau t/2} e^{i\sigma(x \pm t\sqrt{4\sigma^2 gh - \tau^2})} \quad \textbf{(2.7)}$$

and

$$u(x, t) = u^o e^{-\tau t/2} e^{i\sigma(x \pm t\sqrt{4\sigma^2 gh - \tau^2})} \quad \textbf{(2.8)}$$

where ζ^o and u^o are the Fourier coefficients A_n from Equation (2.1).

Computing these solutions for $t = t+\Delta t$ and dividing out the solution at t, gives the per-time-step propagation factor, $\lambda \equiv b(x, t + \Delta t)/b(x, t)$, for the continuum wave as

$$\lambda = e^{-\tau\Delta t/2} e^{\pm i\sigma c\Delta t} \quad \textbf{(2.9)}$$

where the continuum wave speed, or celerity, is expressed as

$$c = \sqrt{gh - (\tau/2\sigma)^2}. \quad \textbf{(2.10)}$$

From Equation (2.9), one can see that in a single time step the wave will decrease in amplitude by a factor of $e^{-\tau\Delta t/2}$ and travel a distance equal to $c\Delta t$.

Given the continuum solution, we now turn to the discrete system and present the Fourier analysis for a general discretization of the SWE. For the fully discretized system, a single component of the Fourier series, represented as

$$b_j^{\,k} = b_0 \mathrm{e}^{i\omega k\Delta t} \mathrm{e}^{i\sigma j\Delta x}, \tag{2.11}$$

where j,k are the spatial and temporal discretization indices, respectively, and b_0 is the amplitude, can be substituted into the discrete representation of Equations (1.5), (1.6), and (1.7) for each independent variable. The discrete per-time-step propagation factor is defined as:

$$\lambda' \equiv \frac{b_j^{\,k+1}}{b_j^{\,k}} = \mathrm{e}^{i\omega\Delta t} \tag{2.12}$$

where the prime differentiates the discrete propagation factor from the continuum factor. Definitions (2.11) and (2.12) are substituted into the set of discrete equations for the algorithm under study and a new system results;

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} \zeta_0 \\ u_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{2.13}$$

where the form of the matrix coefficients will vary depending upon the spatial and temporal discretization. A nontrivial solution will exist only when the determinant of the coefficient matrix is zero, which results in a polynomial in the discrete per-time-step propagation factor λ', whose complex roots can then be written as a function of the wave number, σ.

The stability of the numerical algorithm can be assessed by computing $|\lambda'|$ and verifying that its magnitude is less than or equal to 1 for all desired values of σ. Meanwhile,

damping and phase properties of the numerical solution, relative to the continuum solution, are determined by computing the complex (per-wavelength) propagation factor T, which is defined as

$$T \equiv \left(\frac{\lambda'}{\lambda}\right)^{N_n} \tag{2.14}$$

where N_n is the number of time steps required for the continuum wave to propagate exactly one wavelength ($N_n = L_n/c\Delta t$).

The magnitude of T is the ratio of the amplitude of the discrete wave to the amplitude of the continuum wave after propagating N_n time steps. If $|T|$ is greater than 1, the discrete wave has greater amplitude than the analytical wave and the solution is underdamped, but it is not necessarily unstable, since T is not simply a measure of the discrete wave itself but is a comparison to the continuum wave. While if $|T|$ is less than 1, the discrete wave has smaller amplitude than the analytical wave and the solution is damped.

The argument, or angle, of T is the phase of the discrete wave compared to the analytical wave, where a positive phase indicates lead and a negative phase indicates lag. For a numerical algorithm to perfectly match the continuum solution of the model problem, the magnitude of T should equal 1.0 and the phase of T should equal 0.0 for all components of the solution.

2.2.2 Dispersion analysis

Again, this analysis tool utilizes the assumption that separation of variables can be used to express the solution of the differential equations as periodic in time and space. However, it differs from Fourier analysis in that the time variable remains continuous. The analysis results in a relationship between the temporal frequency and the wave number, often referred to as a dispersion relation, as opposed to the propagation factor in Fourier

analysis. In practice, this dispersion relation can be thought of as an intermediate step in the continuum solution obtained in Fourier analysis. It is this relation that allows one to write the per-time-step propagation factor, Equation (2.9), without explicit dependency on ω.

Platzman [1981] was the first to apply dispersion analysis to the shallow water equations; he demonstrated that a monotonic curve indicates a numerical solution free from spurious $2\Delta x$ oscillations. A folded curve indicates aliasing of wave components, with one wave corresponding to the long physical wave while the other corresponds to short-wavelength noise in the solution. *Kinnmark* [1986] has since demonstrated that for linear elements, but not for quadratic or other elements, a folded curve is a necessary and sufficient condition for the appearance of spurious modes in the numerical solution of the SWE. However, a non-folded curve does not necessarily imply that the solution will be free from these spurious modes. *Atkinson et al.* [2004] have shown that a non-folded dispersion curve in 1D is not always indicative of 2D simulation behavior in that a scheme which exhibits a non-folded curve in 1D analysis can exhibit a folded curve in 2D analysis. He also shows that dispersion curves in 2D spaces are highly dependent upon the grid scheme and the direction that the wave is propagating relative to the grid (i.e., the dispersion surface in 2D may be monotonic for some wave directions and folded for others). However, his analysis did verify that a folded dispersion curve in 1D is a necessary and sufficient condition for a folded dispersion curve in 2D. Therefore, as a first step, it is instructive to examine the 1D dispersion relationships for new algorithms before proceeding with the analysis in higher dimensions (i.e., a folded curve in 1D indicates that further study is not warranted).

The starting point of dispersion analysis is to write the harmonic form of each equation by substituting the temporal harmonic components expressed as

$$\zeta = \hat{\zeta} e^{i\omega t} \quad \text{and} \quad u = \hat{u} e^{i\omega t} \qquad \textbf{(2.15)}$$

into the differential equations. Here $\hat{\zeta}$ is the spatial harmonic of the elevation and $\hat{u}$ is the spatial harmonic of the velocity. The continuum dispersion relation can be derived by substituting the continuum spatial harmonics (Fourier components)

$$\hat{\zeta} = \zeta_0 e^{i\sigma x} \quad \text{and} \quad \hat{u} = u_0 e^{i\sigma x} \tag{2.16}$$

into the resulting ordinary differential equations. The solution of the resulting system of equations for the magnitudes u_0 and ζ_0, which proceeds as in Fourier analysis, determines the relationship between the temporal frequency, ω, and the spatial frequency, σ. The magnitude of ω is then plotted against σ to develop the dispersion curve. These plots can be represented either in physical or dimensionless variables; the former is adopted herein.

For the continuum equations, the above analysis steps result in the following system of equations

$$i\omega\zeta_0 + i\sigma h u_0 = 0 \quad \text{and} \tag{2.17}$$

$$i\omega u_0 + \tau u_0 + i\sigma g \zeta_0 = 0 \tag{2.18}$$

which has the dispersion equation

$$\omega = \frac{1}{2}[i\tau \pm \sqrt{4\sigma^2 gh - \tau^2}]. \tag{2.19}$$

These equations are exactly the same as those derived in the continuum Fourier analysis, since the exponential factors divide out in both analyses. Also, note that the dispersion relationship is nothing more than the eigenvalues of the Fourier analysis given in Equation (2.6). This connection between Fourier and dispersion analyses in the continuum equations can be used to relate the numerically generated Fourier and dispersion results, as will be seen in §2.3.2.

For the discrete equations, the dispersion characteristics can be determined by

substituting the discrete harmonic spatial solutions, expressed as

$$\hat{\zeta} = \zeta_0 e^{ij\sigma\Delta x} \text{ and } \hat{u} = u_0 e^{ij\sigma\Delta x} \quad \textbf{(2.20)}$$

into the spatially discretized ordinary differential equations, where the temporal component has already been treated in the same way as for the continuum dispersion relationship. Here also $i = \sqrt{-1}$, j is the spatial node index, σ is the spatial frequency of the solution, Δx is the discrete spacing of nodes, ζ_0 is the magnitude of the elevation solution, and u_0 is the magnitude of the velocity solution.

Once Equations (2.20) are substituted into the discrete equations, the analysis proceeds as for the continuum equations. The result is a polynomial expression for the discrete temporal frequency, ω', as a function of σ, which is then solved for its complex roots. (As in the Fourier analysis above, the prime notation is introduced to differentiate between the continuum dispersion relationship and the discrete relationships.) The magnitude of the roots *vs.* wave number is plotted as a dispersion curve. For more background information on dispersion analysis as applied to the SWE see *Foreman* [1983, 1984], *Platzman* [1981] and *Walters* [1983].

2.2.3 Analytical propagation relationships for the SWE

In this section the staggered finite difference (SFD) scheme of *Leendertse* [1967], the leap-frog primitive finite element (PLF) method, the selective lumping finite element (SLFE) method of *Kawahara et al.* [1982], the generalized wave continuity (GWC) approach of *Kinnmark* [1986], the low-order finite volume method (FVM), and the low-order discontinuous Galerkin (DG) method are examined. (The higher-order FVM and DG methods are the subject of §3.6.1 and §3.6.2.) A comparison matrix of the discretization features for these algorithms is provided below in Table 2.1. Unless otherwise noted, the discretization features apply to both the continuity and momentum equations. Complete algorithm derivations are provided in Appendices A and B.

Table 2.1 Discretization features in time and space for study algorithms.

	Temporal discretization	Spatial interpolates	Spatial Continuity
SFD	Crank-Nicolson	N/A (centered finite difference)	continuous
PLF	leap-frog	piecewise linear	continuous
SLFE	two-step explicit	piecewise linear	continuous
GWC	three-level centered for Eq.(1.7) two-level centered for Eq.(1.6)	piecewise linear	continuous
Low-order FVM or DG	one-step explicit	piecewise constant	discontinuous

The per-time-step propagation factor λ' and the dispersion relation that result from the analytical treatment of the discretization schemes, which were presented in §2.2.1 and §2.2.2, are summarized in Tables 2.2 and 2.3. The complete derivation of each algorithm's propagation properties are provided in Appendix C. The continuum propagation factor is as in Equation (2.9), and the continuum dispersion relation is as in Equation (2.19). Recall that the discrete phase characteristics are distinguished from the continuum characteristics with the prime notation, e.g., λ' versus λ.

In Table 2.2, note that the GWC formulation results in three roots due to the second order time derivative. In the one-dimensional context, two of these are physical roots (backward and forward traveling waves), while the third is a numerical artifact of the time differentiation. In the interest of brevity, only the cubic polynomial in λ' is reported instead of the actual roots. For the SFD algorithm, θ is the time weighting parameter, which equals 1/2 for the Crank-Nicolson scheme. Also, for the SLFE algorithm, *e* is the selective lumping parameter from Kawahara's scheme and *f* is the selective lumping parameter for the bottom friction term, which has been added to allow independent study of this term. These parameters take on values between 0, unlumped, and 1, fully lumped.

For the dispersion relationships in Table 2.3, the GWC algorithm again results in three roots, where only two correspond to physical waves. As in the previous table, only

Table 2.2 Per-time-step propagation factors from analytical Fourier analysis (derivations in Appendix C).

Continuum	$\lambda = \exp(-\tau\Delta t/2)\exp(\pm i\sigma c\Delta t)$
SFD	$\lambda' = 1 + \frac{2S\theta - \tau\Delta t T \pm \sqrt{(\tau\Delta t T)^2 + 4ST}}{2(T(1+\tau\Delta t\theta) - S\theta^2)}$
PFE	$\lambda' = \sqrt{(1-\tau\Delta t) + \frac{K}{2} \pm \sqrt{\frac{9\mu^2 B^2}{(A+4)^2}K - (\tau\Delta t - 1)K + (\tau\Delta t)^2}}$
SLFE	$\lambda' = \frac{1}{2}[2E - \tau\Delta t J \pm \sqrt{(\tau\Delta t J)^2 + \mu^2 g h F^2}]$
GWC	$P_3\lambda'^3 + P_2\lambda'^2 + P_1\lambda' + P_0 = 0$
Low-order FVM or DG	$\lambda' = \frac{1}{2}[2 + \mu a(A-2) - \tau\Delta t \pm \sqrt{(\tau\Delta t)^2 + gh(\mu B)^2}]$

$A = 2\cos(\sigma\Delta x)$, $B = 2i\sin(\sigma\Delta x)$, $C = 2\cos(2\sigma\Delta x)$, $D = 2i\sin(2\sigma\Delta x)$

$E = \frac{2+e}{3} + \frac{1-e}{6}A + \frac{\mu^2 gh}{8}(C-2)$, $F = \frac{2+e}{3}B + \frac{1-e}{6}D - \frac{\tau\Delta t}{2}\left[\frac{2+f}{3}B + \frac{1-f}{6}D\right]$,

$J = \frac{1-f}{6}\left[(C+2)\left[\frac{1-e}{6} - \frac{\tau\Delta t}{2}\frac{1-f}{6}\right] + A\left[\frac{2+e}{3} - \tau\Delta t\frac{2+f}{3}\right]\right] + \frac{2+f}{3}\left[A\frac{1-e}{6} + \frac{2+e}{3} - \frac{\tau\Delta t}{2}\frac{2+f}{3}\right]$

$K = (36gh\mu^2B^2)/(A+4)^2$, $S = 2i\sin(\sigma\Delta x/2)g\mu^2 h(e^{i\sigma\Delta x} - 1)$, $T = e^{(i\sigma\Delta x)/2}$

$c = \sqrt{gh - \tau^2/(4\sigma^2)}$, $\mu = \Delta t/\Delta x$

$K_1 = \frac{\Delta x^2}{6\Delta t}$, $K_2 = \frac{\Delta x^2}{3\Delta t^2}$, $K_3 = \frac{\Delta x^2}{3\Delta t^3}$, $K_4 = \frac{gh}{8}B^2(G-\tau)$

$P_0 = -K_3\left[2 + \frac{A}{2}\right] + (K_2(G+\tau) - K_1 G\tau)\left[1 + \frac{A}{4}\right] + \alpha_3 gh\left[1 - \frac{A}{2}\right]\left[\tau - \frac{2}{\Delta t}\right]$

$P_1 = 3K_3\left[2 + \frac{A}{2}\right] - (K_2(G+\tau) + K_1 G\tau)\left[1 + \frac{A}{4}\right] + gh\left[1 - \frac{A}{2}\right]\left[\frac{2}{\Delta t}(\alpha_3 - \alpha_2) + \tau(\alpha_3 + \alpha_2)\right] - K_4$

$P_2 = -3K_3\left[2 + \frac{A}{2}\right] - (K_2(G+\tau) - K_1 G\tau)\left[1 + \frac{A}{4}\right] + gh\left[1 - \frac{A}{2}\right]\left[\frac{2}{\Delta t}(\alpha_2 - \alpha_1) + \tau(\alpha_2 + \alpha_1)\right] - K_4$

$P_3 = K_3\left[2 + \frac{A}{2}\right] + (K_2(G+\tau) + K_1 G\tau)\left[1 + \frac{A}{4}\right] + \alpha_1 gh\left[1 - \frac{A}{2}\right]\left[\tau + \frac{2}{\Delta t}\right]$

the cubic polynomial in ω' is reported. Note also that a dispersion relation cannot be derived for the SLFE algorithm due to the two-step temporal discretization (i.e., it is impossible to write the algorithm in a time-independent form). The coefficients *A* and *B*, which appear in this table, are the same as reported in Table 2.2.

Table 2.3 Analytical dispersion relationships for study algorithms (derivations in Appendix C).

Continuum	$\omega = \frac{1}{2}[i\tau \pm \sqrt{4gh\sigma^2 - \tau^2}]$
SFD	$\omega' = \frac{i}{2}[\tau \pm \sqrt{\tau^2 + 4gh(A-2)/\Delta x^2}]$
PFE	$\omega' = \frac{i}{2}\left[\tau \pm \sqrt{\tau^2 + \frac{36gh}{\Delta x^2}\left[\frac{B}{3A+2B}\right]^2}\right]$
GWC	$\left[\frac{-i}{6}(4+A)\right]\omega'^3 + \left[\frac{-i}{6}(G+\tau)(4+A)\right]\omega'^2 + \left[\frac{i}{6}G\tau(4+A) + \frac{ihg}{\Delta x^2}(2-A)\right]\omega' + \left[\frac{hg\tau}{\Delta x^2}(2-A) + \frac{hg}{4\Delta x^2}(\tau - G)B^2\right] = 0$
Low-order FVM / DG	$\omega' = \frac{i}{2}[\tau - a(A-2)/\Delta x \pm \sqrt{\tau^2 + (aB/\Delta x)^2}]$

2.3 Numerical Propagation Analysis Tools

While the above analytical approaches are widely used in many fields to analyze the behavior of differential and difference equations, they have not been frequently employed in numerical analyses for systems that do not allow such an analytical approach. For instance, the FVM or DG algorithms with a high-order discretization do not lend themselves to analytical analysis, since the slope limiting schemes result in an algorithm that evolves in time and cannot be written in a closed-form. Many standard limiters and hybrid variations on these exist, but currently, there is no clear method for choosing the "best" slope limiter. Often, this choice is based on a trial-and-error process, where the first successful limiter is used. In an effort to systematically study the properties of some of the more popular limiters as applied to the FVM method, numerical analogs to traditional Fourier and dispersion analyses were developed and will be presented herein.

Past efforts in this area include the work of *Kolar* [1992] who developed a numerical analysis technique analogous to the above analytical Fourier analysis for the 1D

linear advection equation. However, some modification is necessary to apply the same technique to a system of equations, such as the SWE. The work of *Vichnevetsky and Shieh* [1972], who applied a related procedure to the scalar transport equations, is also acknowledged. *Kinnmark and Gray* [1985] developed a "2Δx-test" for the specific examination of spurious 2Δx waves, but it is desirable to examine the propagation characteristics of the full spectrum. Therefore the following procedure was developed.

The general theory behind the following numerical technique is to use the numerical discretization scheme under study to propagate a single wave with known amplitude and wavelength exactly one wavelength forward in time and then examine the final location and amplitude of the wave. This information is then translated into phase and damping characteristics (for the equivalent Fourier analysis) and dispersion characteristics (for the equivalent dispersion analysis). The procedure is then repeated for each wavelength with the final product being numerically generated phase error, damping, and dispersion curves. In this way, the propagation characteristics of the linear algorithm can be obtained one wavelength at a time.

As mentioned in the introduction, advantages of such an approach are the ability to study full discrete schemes, including boundary effects, the generality of working without a closed-form equation, and a direct relationship between the propagation behavior and actual simulation output. For simplification only domains of constant bathymetric depth are considered in this study; the general procedure should be applicable to more complicated bathymetric domains as well, but the wave celerity would not be constant over the domain and the formula for calculating the number of time steps to propagate the wave exactly one wavelength would be more complicated.

A schematic representation of the general procedure is shown below in Figure 2.1, where the illustrated curves were generated for the low-order FVM algorithm using the center initialization method and the 4-step procedure that will be explained in the next

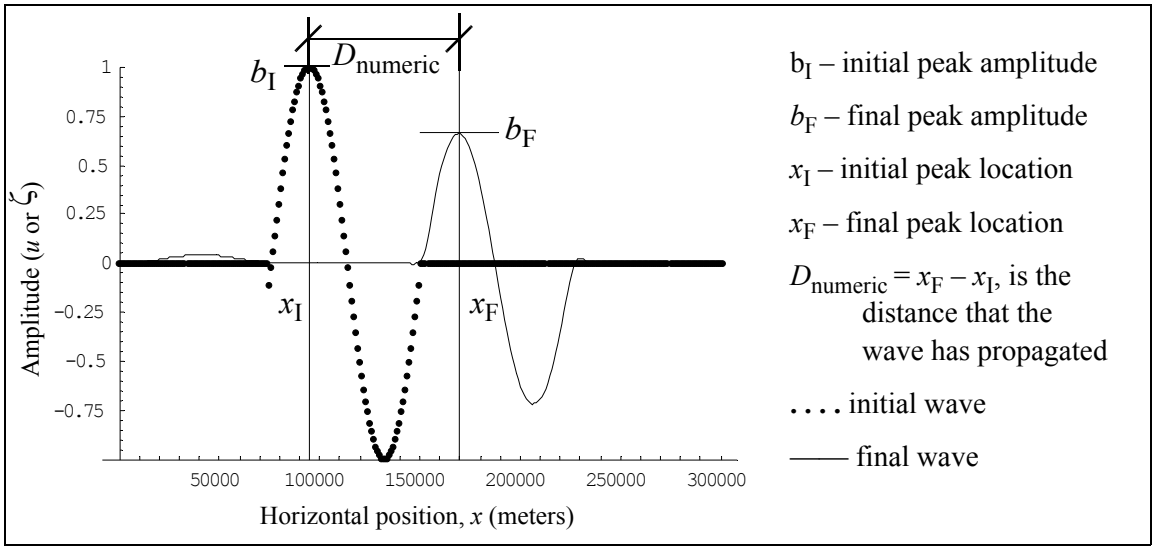


Figure 2.1 Schematic of peak wave tracking technique used in numerical analogs to Fourier and dispersion analyses ($L_n/\Delta x = 75$).

section. The first wave (discrete points) is the initial wave introduced into the center of the domain at time, t=0, and the second wave (solid line) is the propagated wave after N_n' time steps of length Δt. Note that the initial wave has been shifted such that the peak lies directly on a node point (to facilitate wave identification). Correspondingly, note that the final peak does not, in general, fall directly on a grid point (a further consequence of discrete numerical schemes representing continuous waves). In this study, the local maximum value from the discrete grid points will be used as an approximation for the peak instead of using interpolation to estimate the peak location. Note also that the trailing portion of the second wave is not identically zero and that there is some wiggle in the leading portion as well; this is an indication of the inherent problems in numerical discretizations. For reference, the notation used in this figure will be carried throughout the following derivations.

As mentioned above, there is some error introduced by not knowing the exact location of the final peak, since it most likely lies between grid points. In order to characterize these errors, two other methods of tracking the wave position; namely, a center of mass (COM) approach, by which the initial and final waves are numerically integrated

and first moment principles are used to determine the "center" of the wave, and a least squares harmonic (LSH) analysis are examined. For the LSH analysis the final waveform is "fit" with a wave of the appropriate wavelength and the amplitude and phase shift are determined in a least squares sense. These wave tracking methods will be discussed in more detail in *Step 3* of the following section.

2.3.1 Numerical Fourier analysis

Step 1: Initialize with a single wave of known amplitude, wavelength, and position.

In general, there are two initialization options available for introducing the wave into the domain: 1) an initial value problem (IVP) where one full wavelength is used as the initial condition at t=0, or 2) a boundary value problem (BVP) where the open ocean boundary condition is used to force the domain with the appropriate wavelength for one full period.

In developing the procedure for numerical propagation, it was not clear at the outset which of these two options for introducing the wave into the domain was best. Thus, qualitative and quantitative error measures are examined herein to determine the importance of the initialization procedure and any possible boundary effects on the final phase results. The error measures will be discussed when the numerical propagation techniques are validated with analytical propagation results in §2.4. The following methods for initializing the starting wave are examined:

Method 1. Initialize wave at the open ocean boundary

Method 2. Initialize wave near the center of the domain

Method 3. Use elevation forcing of appropriate wavelength (period) at the open ocean boundary to bring the wave into the domain.

For Methods 1 and 2, the elevation and velocity profiles are identically initialized with one wavelength of the wave at time t=0 and then the waves are allowed to propagate for one

full period. For Method 3, the open ocean boundary condition is used to force the system with a wave of the desired wavelength for one full period, then the forcing is turned off and the wave is allowed to propagate for another full period.

The domain size depends upon the wavelength being simulated and must be large enough that the wave will not run into the land boundary (and reflect back into the domain). A domain of 3 wavelengths is sufficient when the wave is initialized at the open boundary, either through an IVP or a BVP, and 4 wavelengths is sufficient if the wave is initialized near the center of the domain. For Methods 1 and 2, the wave may be shifted, such that the initial peak falls on a grid point, for ease of comparison. In this study, one wavelength of a sine wave represented as,

$$\sin\sigma(x-\phi) \;=\; \sin\frac{2\pi(x-\phi)}{L_n}, \tag{2.21}$$

where ϕ is an appropriate shift, is used to define the initial wave for Methods 1 and 2. Only $L_n + 1$ grid points are initialized, such that a single wave is within the domain, and the remaining grid points are set to zero. Notice in Figure 2.1 that with Method 2 the wave is initialized from $x = L_n$ to $x = 2L_n + 1$ and the wave does not start at zero, but has been shifted so that the peak is on a grid point.

For Method 3, the desired wavelength is related to the period by $\mathrm{T} = L_n / c$ (where the period, T, should not be confused with the complex propagation factor, T) and the elevation is forced at the open boundary by a sine wave represented as

$$\sin\frac{2\pi t}{\mathrm{T}} \tag{2.22}$$

where t is the current simulation time. The simulation is run for two full periods, the first with boundary forcing to introduce the wave and the second with no forcing to propagate the initial wave, for a total of $2N_n$ time steps. For this initialization method, one of the wave

tracking methods discussed in *Step 3* must be used to determine the initial location and amplitude of the wave. For consistency, the same wave tracking method will be used to analyze the initial wave as is used for the final wave.

Step 2: Execute SWE algorithm and propagate wave for N_n' timesteps.

Once the wave has been introduced to the domain, the next step is to propagate the wave forward approximately one wavelength by running the SWE solution algorithm for enough time steps so that the wave would travel one wavelength analytically. From the analytical Fourier analysis in §2.2.1, it is known that the wave should travel exactly one wavelength in $N_n = L_n/(c\Delta t)$ time steps where the wave celerity, c, depends upon the wavelength. Of course, within the framework of any computational procedure, the computer cannot compute the results for fractional time steps and the mathematical ceiling (or floor) of N_n must be used for the number of time steps in the actual code. Although this does not introduce a large error into the numerical phase scheme, it is easily corrected in the analytical wave solution and should be taken into consideration before computing the complex propagation factor T.

In order to account for this additional propagation time in the analytical results, the corrected propagation time is defined as N_n' = ceiling[N_n] such that the analytical solution predicts a final amplitude of

$$|\lambda| = e^{-\tau N_n' \Delta t/2} \tag{2.23}$$

and a distance travelled of

$$D_{\text{analytic}} = N_n' c \Delta t \tag{2.24}$$

for an initial wave of amplitude 1. Using N_n' instead of N_n in Equation (2.14) allows an equal comparison between the numerical and analytical results.

Step 3: Compute the final position and amplitude and compare to the initial wave from Step 1.

For initialization Methods 1 and 2 from *Step 1*, the exact location and magnitude of the initial peak are known. However, determining the location of the initial peak for initialization Method 3 and the final peak for all three methods is somewhat more difficult, since the peaks will most generally not lie on a grid point. Herein, three "wave tracking" techniques are examined: peak tracking, center of mass tracking, and least squares harmonic reconstruction of the discrete data. The determination of the propagation properties of an algorithm using each of these tracking techniques are described in the next three subsections. Since the wave has moved forward approximately one wavelength, the propagation properties from each tracking technique are most naturally reported on a per wavelength basis, which also allows comparison with the complex propagation factors computed in §2.2.3.

Peak tracking

In this wave tracking technique the local maximum from the discrete data is used as an approximation for the peak, rather than using interpolation to determine the exact peak location and magnitude. For the well-resolved, larger physical wavelengths, little error arises from not knowing the exact location of the peak. Referring to Figure 2.1, the propagation properties for each wavelength are then computed as

$$|T| = (b_F / b_I) / |\lambda| \tag{2.25}$$

and

$$\angle T = (D_{\text{numeric}} - D_{\text{analytic}}) 2\pi / L_n \tag{2.26}$$

where $|\lambda|$ is as given in Equation (2.23), b_I is the initial peak amplitude, b_F is the final peak amplitude for u or ζ (approximated by the local maximum as discussed above) after the

wave travels N_n' time steps, $\angle T$ is the argument or phase of T, and the numeric distance traveled is quantified as

$$D_{\text{numeric}} = x_{\text{F}} - x_{\text{I}} \tag{2.27}$$

where x_{I} and x_{F} are the initial and approximate final peak locations. From the expression for the phase, it is apparent that if the change in peak position is exactly equal to the analytic distance, then the phase error is zero; and if the peak travels more (less) than the analytic distance, then there is a phase lead (lag). Similarly, if the final peak amplitude is greater (less) than that of the analytic wave, then the numeric wave is amplified (damped) relative to the analytic solution.

Similarly, one can examine the relative wave speed or celerity of the numerically propagated wave by computing the ratio of distances traveled in N_n' time steps, since the time is the same for both the numerical and analytical waves. This ratio, represented as

$$\frac{c'}{c} = \frac{D_{\text{numeric}}}{D_{\text{analytic}}} \tag{2.28}$$

where c' is the numerical wave speed, is also an indicator of the phase properties. When the numerical wave speed is equal to the analytical wave speed, indicated by a ratio of 1.0, the wave has propagated at the same speed as predicted analytically, and there will be no phase error. In the same way, a ratio greater (less) than 1.0 would indicate a phase lead (lag) in the solution.

Center of mass

For this wave tracking technique, both the initial and final waves are integrated numerically to determine their respective "mass" and "center of mass." In comparing the initial and final waves, the ratio of final to initial mass is taken to be an approximation of the damping of the algorithm, and the change in position of the center of mass is taken as

an indication of the phase. Figure 2.2 gives a general schematic of the notation and calculation procedures that are used in this tracking technique, as well as the results for a 75Δx wave from the FVM algorithm.

For the example, notice that the initial wave lies entirely within a single wavelength, but that the final wave goes outside of the solid vertical lines, which indicate where the final wave should be located. Although the final wave is symmetric, so that the center of mass is located correctly, the total mass will be higher than if the wave fit within the solid lines, and the estimate for the damping ratio will be higher than it should be as a result of this.

The trapezoidal rule for numerical integration is used to calculate the total mass and first moment for each wave. The total mass is calculated as

$$mass = \sum_{i=1}^{nn-1} \left[(x_{i+1} - x_i) \frac{|b_i + b_{i+1}|}{2} \right] \tag{2.29}$$

where *nn* is the number of discrete data points in the domain, and the total moment is

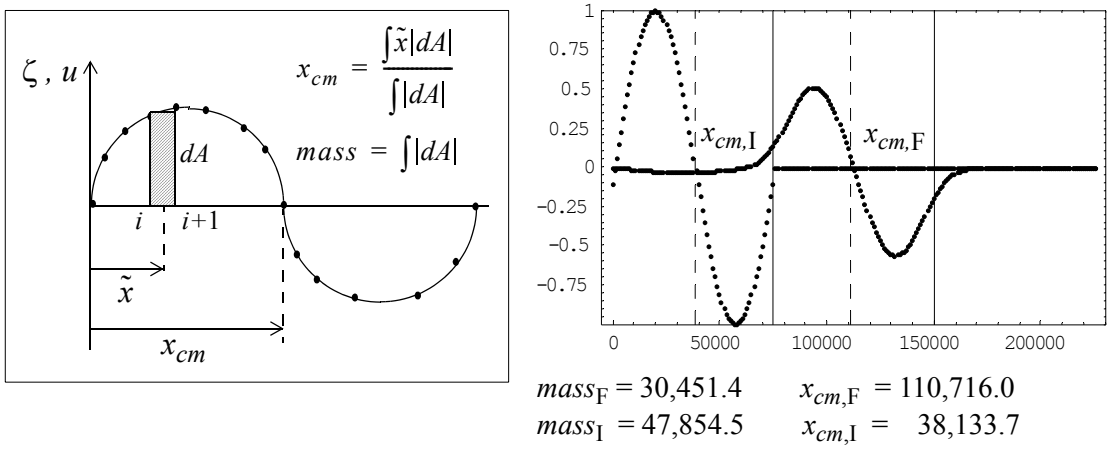


Figure 2.2 Notation for center of mass wave tracking technique and example results from 75Δx wave using the FVM algorithm.

calculated as

$$moment = \sum_{i=1}^{nn-1} \left[\frac{(x_i + x_{i+1})}{2} (x_{i+1} - x_i) \frac{|b_i + b_{i+1}|}{2} \right] \tag{2.30}$$

where the notation is as in Figure 2.2. The center of mass, x_{cm}, for each wave is then calculated as

$$x_{cm} = \frac{moment}{mass}. \tag{2.31}$$

Notice that the absolute value of the discrete wave data is used for the integrations to avoid returning a zero mass or moment.

The Fourier propagation properties for each wavelength are then computed as

$$|T| = (mass_F / mass_I) / |\lambda| \tag{2.32}$$

and

$$\angle T = (D_{\text{numeric}} - D_{\text{analytic}}) 2\pi / L_n \tag{2.33}$$

where $|\lambda|$ is as given in Equation (2.23), $mass_I$ and $mass_F$ are the masses of the initial and final waves (either u or ζ) as calculated by Equation (2.29), and the numeric distance traveled is quantified as

$$D_{\text{numeric}} = x_{cm,\,F} - x_{cm,\,I} \tag{2.34}$$

where $x_{cm,\,F}$ and $x_{cm,\,I}$ are the centers of mass for the final and initial waves, respectively.

As with the peak tracking technique, it is apparent that if the change in center of mass is exactly equal to the analytic distance, then the phase error is zero; and if the center of mass travels more (less) than the analytic distance, then there is a phase lead (lag).

Similarly, if the final to initial mass ratio is greater (less) than the amplitude of the analytic wave, then the numeric wave is amplified (damped) relative to the analytic solution.

Least squares harmonic (LSH) fit

For this wave tracking technique, the propagated numerical wave is fit by a general cosine wave given as

$$f(x) = A_0 + C_1 \cos(\sigma x + \theta) \qquad \textbf{(2.35)}$$

where A_0 is the mean value, or the average height above the abscissa, C_1 is the amplitude of the wave, σ is the wave number and is equal to $2\pi/L_n$, and θ is the phase shift relative to where the cosine wave begins a new cycle. Figure 2.3 shows the LSH fit of the $75\Delta x$ wave for the FVM and gives the general notation of the data points and the fitting wave. The initial wave has an exact fit with $C_1 = 1.0$, $A_0 = 0.0$ and $\theta = 0.0$, since it is initialized with a perfect sinusoid, and the fit parameters for the final wave are shown in the figure.

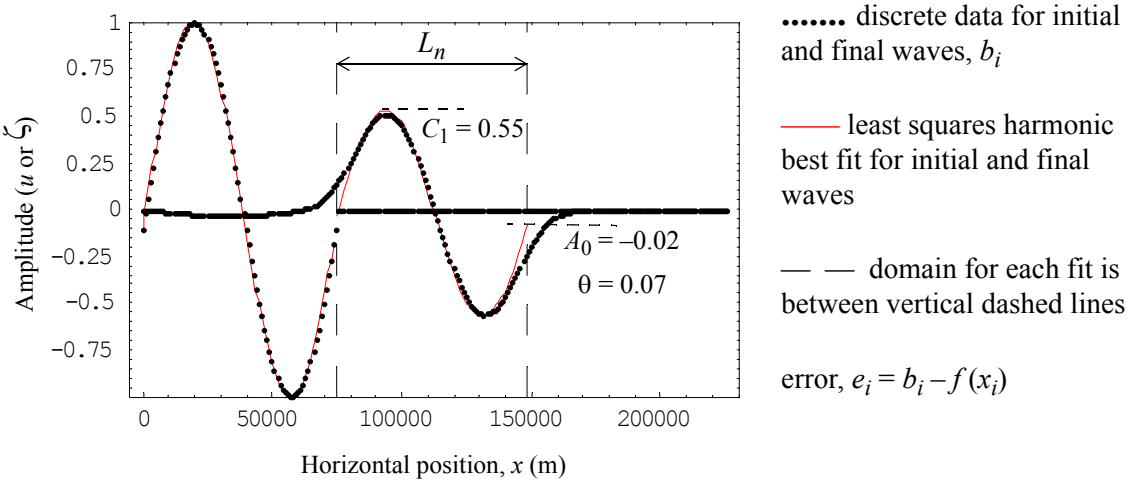


Figure 2.3 Notation for least squares harmonic fit wave tracking technique ($L_n = 75\Delta x$).

The data points, b_i, can be either elevation or velocity data and the solid red curves are the sine waves to which the data is being fit. The wave is fit in a least squares sense, where the errors between the best-fit sinusoid and the actual data are minimized. For such a curve fitting application, Equation (2.35) can be rewritten in a more useful form by using the trigonometric identity

$$C_1\cos(\sigma x+\theta) = C_1[\cos(\sigma x)\cos(\theta)-\sin(\sigma x)\sin(\theta)] \quad \textbf{(2.36)}$$

which gives, after substitution,

$$f(x) = A_0 + A_1\cos(\sigma x) + B_1\sin(\sigma x) \quad \textbf{(2.37)}$$

where

$$A_1 = C_1\cos\theta \text{ and } B_1 = -C_1\sin\theta. \quad \textbf{(2.38)}$$

Next, the error at each data point is defined to be

$$e_i = b_i - f(x_i) = b_i - A_0 - A_1\cos(\sigma x_i) - B_1\sin(\sigma x_i) \quad \textbf{(2.39)}$$

and the square residual is given as

$$S_r = \sum_i e_i^2 = \sum_i [b_i - A_0 - A_1\cos(\sigma x_i) - B_1\sin(\sigma x_i)]^2, \quad \textbf{(2.40)}$$

where i is only summed over the data points which lie within the domain of the assumed wave; one wavelength or k data points, where $L_n = k\Delta x$ and the data points are equally spaced. The residual is minimized relative to the coefficients of the assumed cosine wave to get

$$\frac{\partial S_r}{\partial A_0} = \sum_i [2A_0 - 2b_i + 2A_1\cos(\sigma x_i) + 2B_1\sin(\sigma x_i)] \equiv 0 \quad \textbf{(2.41)}$$

$$\frac{\partial S_r}{\partial A_1} = \sum_i [2(A_0 - b_i)\cos(\sigma x_i) + 2A_1\cos(\sigma x_i)^2 + 2B_1\sin(\sigma x_i)\cos(\sigma x_i)] \equiv 0 \qquad \textbf{(2.42)}$$

$$\frac{\partial S_r}{\partial B_1} = \sum_i [2(A_0 - b_i)\sin(\sigma x_i) + 2A_1\cos(\sigma x_i)\sin(\sigma x_i) + 2B_1\sin(\sigma x_i)^2] \equiv 0\,, \qquad \textbf{(2.43)}$$

which gives the following system of equations to solve for the three coefficients:

$$\begin{bmatrix} k & \sum\cos(\sigma x_i) & \sum\sin(\sigma x_i) \\ \sum\cos(\sigma x_i) & \sum\cos(\sigma x_i)^2 & \sum\cos(\sigma x_i)\sin(\sigma x_i) \\ \sum\sin(\sigma x_i) & \sum\cos(\sigma x_i)\sin(\sigma x_i) & \sum\sin(\sigma x_i)^2 \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \\ B_1 \end{bmatrix} = \begin{bmatrix} \sum b_i \\ \sum b_i\cos(\sigma x_i) \\ \sum b_i\sin(\sigma x_i) \end{bmatrix} \qquad \textbf{(2.44)}$$

where the summations are assumed to be over k data points.

When the sample data is equally spaced, as it is when Δx is constant, this system can be simplified to get

$$A_0 = \frac{1}{k}\sum b_i \qquad \textbf{(2.45)}$$

$$A_1 = \frac{2}{k}\sum b_i\cos(\sigma x_i) \qquad \textbf{(2.46)}$$

$$B_1 = \frac{2}{k}\sum b_i\sin(\sigma x_i) \qquad \textbf{(2.47)}$$

Now the relationships in Equation (2.38) are used with Equations (2.46) and (2.47) to get the best fit amplitude and phase shift of the data points as

$$C_1 = \sqrt{A_1^2 + B_1^2} \qquad \textbf{(2.48)}$$

$$\theta = \begin{cases} \operatorname{atan}(-B_1/A_1) & \text{if } A_1 \geq 0 \\ \operatorname{atan}(-B_1/A_1) + \pi & \text{if } A_1 < 0 \end{cases} \tag{2.49}$$

The above derivation assumes that the data is being fit with a cosine wave. Since the initial wave in *Step 1* is generated using a sine wave, it is necessary to convert the phase shift given by Equation (2.49) to the equivalent shift in a sine wave by adding $\pi/2$ which gives

$$\theta = \begin{cases} \operatorname{atan}(-B_1/A_1) + \dfrac{\pi}{2} & \text{if } A_1 \geq 0 \\ \operatorname{atan}(-B_1/A_1) + \dfrac{3\pi}{2} & \text{if } A_1 < 0 \end{cases} \tag{2.50}$$

for the final shift.

When the initial wave from *Step 1* is generated using Method 1 or 2, the peak location and amplitude are known and the Fourier propagation characteristics of the algorithm can be given as

$$|T| = (C_1/b_{\mathrm{I}})/|\lambda| \text{ and } \angle T = \theta - \sigma\phi \tag{2.51}$$

where b_{I} is the amplitude of the initial peak, σ in the wavenumber, ϕ is the shift from the initialization, and the domain for the curve fitting depends upon the initialization method. For Method 1, where the wave is initialized at the boundary between $i = 1$ and $i = k + 1$, the final wave should be between $i = k + 1$ and $i = 2k$; and for Method 2, where the wave is initialized between $i = k + 1$ and $i = 2k + 1$, the final wave should be between $i = 2k + 1$ and $i = 3k$.

When the initial wave is generated using Method 3, the properties of the initial wave are not known exactly and the amplitude and shift of the initial wave are also calculated using a least squares harmonic fit, where the domain of interest is between $i = 1$ and

$i = k$. Then the magnitude and phase of the final wave are compared to the properties of the initial wave before comparing it to the continuum wave to get

$$|T| = (C_1 / C_{1,\,init}) / |\lambda| \text{ and } \angle T = \theta - \theta_{init} \qquad \textbf{(2.52)}$$

where $C_{1,\,init}$ and θ_{init} are the amplitude and shift of the initial wave and the domain of interest for the final wave is from $i = k + 1$ to $i = 2k$.

Step 4: Repeat Steps 1-3 over range of wavelengths to generate plots.

Dynamic allocation of vectors allows several wavelengths to be examined by using a simple loop. Propagation data is output for each wavelength and discrete plots of phase and amplitude for the complex propagation factor versus wavelength are generated.

2.3.2 Numerical dispersion analysis

A connection between the propagation phase from Fourier analysis and the dispersion relationship can be established through the definition for phase speed, given as $c = \omega/\sigma$, when one assumes constant bathymetry and bottom friction, and hence constant phase speed. By rearranging this definition and estimating the phase speed from the wave's final position after N_n' time steps, an expression for ω' can be established as

$$\omega' = \sigma c' = \left[\frac{2\pi}{L_n}\right]\left[\frac{x_F - x_I}{N_n' \Delta t}\right]. \qquad \textbf{(2.53)}$$

Thus, a discrete dispersion plot can be generated one wavelength at a time by using the discrete distance traveled, $D_{\text{numeric}} = x_F - x_I$, which was determined by either peak wave tracking or center of mass wave tracking during *Step 3* above.

Additionally, the concept of relative wave speed introduced in §2.3.1, can be used to determine the temporal frequency of the propagated wave as $\omega' = c'\sigma$, where c' is calculated using Equation (2.28), to get

$$\omega' = \sigma c \frac{D_{\text{numeric}}}{D_{\text{analytic}}}. \tag{2.54}$$

From this second expression for ω', it is clear that when c' is equal to c, or D_{numeric} is equal to D_{analytic}, then the analytical definition for phase speed is recovered. This expression for the temporal frequency can also be adjusted in order to use the phase shift from the least squares harmonic fit in *Step 3*. From the definition for phase shift given in Equation (2.26), the shift from the LSH fit is related to the distance traveled by

$$\theta = \sigma(D_{\text{numeric}} - D_{\text{analytic}}), \tag{2.55}$$

which can be rearranged to determine the numerical distance traveled in the simulation results as follows:

$$D_{\text{numeric}} = \frac{\theta}{\sigma} + D_{\text{analytic}} \tag{2.56}$$

where $\sigma = 2\pi/L_n$ is the wave number. Substituting this expression into Equation (2.54) gives the numerical formula for finding the temporal frequency when a least squares harmonic fit is used to find the wave properties:

$$\omega' = \frac{\sigma c}{D_{\text{analytic}}}\left(\frac{\theta}{\sigma} + D_{\text{analytic}}\right) = \sigma c + \frac{\theta c}{D_{\text{analytic}}}. \tag{2.57}$$

2.4 Validation of Numerical Propagation Analysis Tools

A comparison of the analytical propagation characteristics, as computed in §2.2.3, to the numerically generated results from §2.3 is essentially a comparison of the theoretical propagation behavior predicted by the analytical analyses to the actual propagation behavior one might expect in practice. Hence, one should not expect an exact match between the numerical analog and the analytical results (e.g., the latter does not consider boundary effects). Nevertheless, using the methods as described herein, the numerical

analogs are able to predict the general shape of the propagation curves for all of the discretization schemes under study. The parameter values used in this study are summarized below in Table 2.4, where θ is the time weighting parameter for the SFD algorithm and *e* and *f* are the selective lumping coefficients for the SLFE algorithm.

Table 2.4 Parameters used in validation study.

$g = 9.81$ m/s^2	$h = 10$ m	$\Delta t = 1.0$ (50.0) s	$\theta = 0.5$
$\tau = 0.0001$ s^{-1}	$G = 0.001$s^{-1}	$\Delta x = 1000$ m	$e = f = 0.9$

For the SLFE algorithm, the maximum stable time step must be used or the algorithm is over dissipative. Therefore, a time step of 50.0 seconds was used for this algorithm and a time step of 1.0 second was used for all other algorithms. (An examination of the temporal effects for the SLFE will be presented in the next chapter, §3.5.2.) The length of the simulation domain varies according to the wavelength and initialization method, as discussed in §2.3, as do the boundary conditions. For all three initialization methods a "land boundary" is assumed on the right side of the domain, such that $\partial\zeta/\partial x = 0$ and $u = 0$. For initialization methods 1 and 2 (IVP), the left boundary conditions are given by $\zeta = 0$ and $\partial u/\partial x = 0$ for all $t > 0$. Meanwhile, initialization method 3 uses the left boundary to introduce the wave with on open ocean boundary condition given by $\zeta = \hat{\zeta}(t)$ and $\partial u/\partial x = 0$ for $t < nt$ and then the boundary elevation is set to zero from $nt < t < 2nt$.

To validate the numerical analogs to Fourier and dispersion analysis, which were presented above in §2.3, the numerical propagation results were rigorously compared to the analytical characteristics for each of the discretization algorithms in Table 2.1. While the analytical analyses produce one curve for each algorithm characteristic (damping, phase shift and dispersion), the numerical propagation tools produce two curves for each characteristic - one from the elevation data and one from the velocity data. However,

because the elevation and velocity variables are initialized with the same wave (when initialization Method 1 or 2 is used), the final waves are also similar resulting in comparable propagation results. For initialization Method 3, the elevation and velocity propagation results are also nearly identical. Thus, for the purpose of comparison with the analytical propagation characteristics, the average of the elevation and velocity results will be used. All of the qualitative graphs and quantitative errors were developed in this way.

For each of the three initialization methods, the damping, phase and dispersion results were compared for all three wave tracking techniques for a total of 27 comparisons for each algorithm. Both qualitative and quantitative error measures were used to summarize this large data set. Algorithm specific recommendations are made following each algorithm's qualitative analysis and overall recommendations for all algorithms are given in the final section, §2.5.

Qualitatively, the numerically generated propagation characteristics (damping, phase, and dispersion) were compared with plots of the analytical results from §2.2.3 to determine if any of the initialization methods were grossly in error (i.e., did not capture the general shape and trend of the analytical plots). Figures 2.4 through 2.7 present the phase, damping and dispersion plots comparing the initialization and wave tracking methods for each of the algorithms, except for the SLFE algorithm, which only has analytical Fourier characteristics so that no dispersion plots are given. In each graph, ten sets of data are plotted. The analytical propagation characteristic is shown in the solid black line and the three dashing patterns differentiate between the initialization methods: Method 1) initial wave at boundary shown by dotted lines, Method 2) initial wave near center shown by dash-dot lines, and Method 3) boundary forcing to introduce wave shown by dashed lines. Additionally, the three colors differentiate between the wave tracking techniques: 1) peak tracking in red, 2) center of mass tracking in green, and 3) least squares harmonic tracking in blue. Only the quality of the numerically derived propagation characteristics in comparison to the analytical behavior will be discussed in this chapter; the desired

propagation characteristics for each algorithm will be discussed in Chapter 3.

2.4.1 Staggered finite difference

Validation plots of Fourier phase errors and damping ratios and dispersion curves for the SFD algorithm are given in Figure 2.4. For the damping ratio, peak tracking with

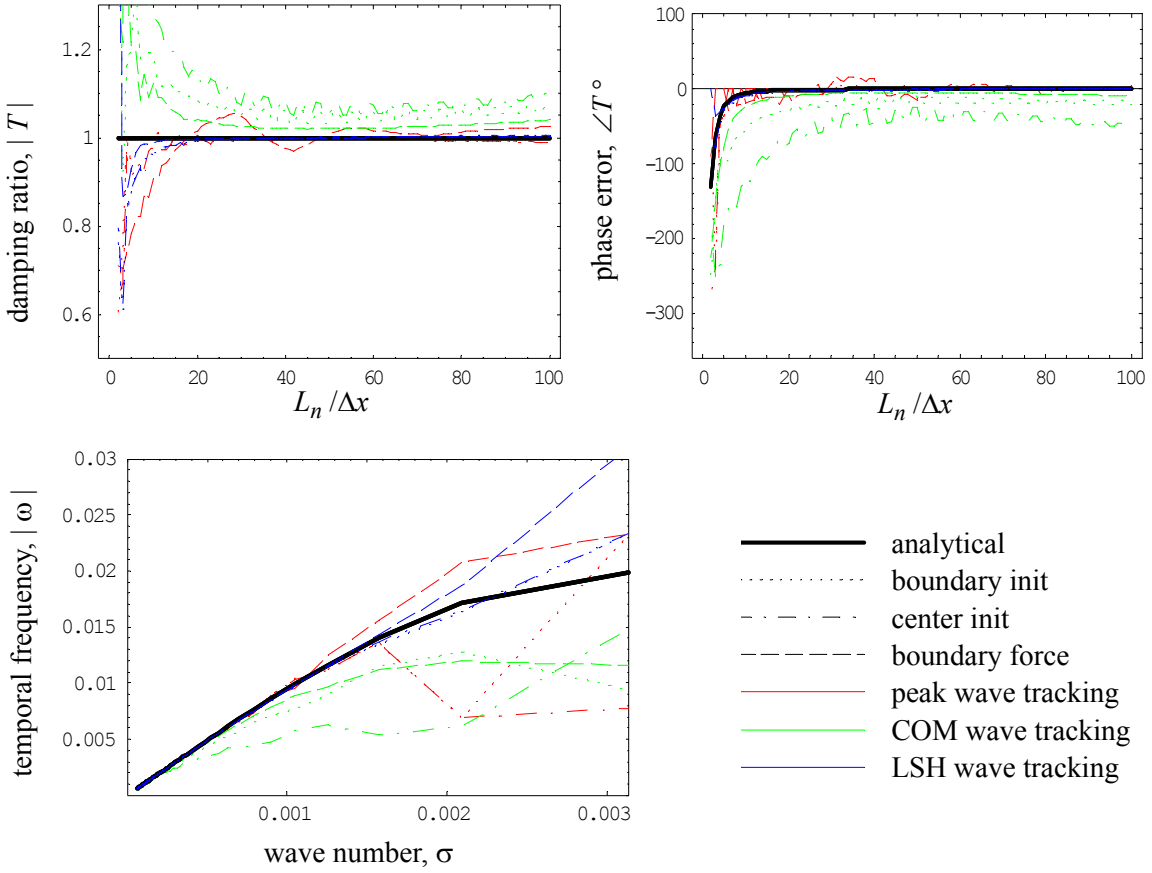


Figure 2.4 Comparison of analytical and numerical propagation properties for the SFD algorithm.

either IVP does a reasonably good job of matching the analytical damping curve, particularly for wavelengths greater than 10Δx. However, the peak tracking for the BVP oscillates about the analytical curve and is not as good of a fit. The COM wave tracking technique does not capture the general trend for any of the initialization methods; but the

LSH wave tracking captures the behavior well, particularly for wavelengths greater than 10Δx. Notice that the boundary forcing LSH curve (blue dashed) shoots up at the 2Δx wavelength; this is due to the best fit of the initial wave. Recall in Equation (2.52) that the damping ratio is approximated by the ratio of the final amplitude over the initial amplitude to the analytical amplitude. For boundary forcing, the initial wave for the 2Δx wavelength has an amplitude close to zero so that the ratio becomes very large even though the final amplitude is also very small.

For the phase error, all of the peak tracking/initialization combinations do a good job, but the IVP initializations match better than the BVP, which oscillates somewhat. The BVP initialization for the COM wave tracking adequately captures the phase behavior, but neither of the IVP methods match well. All of the LSH wave tracking/initialization combinations match the analytical phase behavior almost exactly.

For the dispersion curves, all of the initialization/wave tracking combinations match for the large wavelengths; however, the COM schemes deviate from the analytical curve most quickly as the wavelengths become smaller and, overall, do not predict the correct behavior. The boundary forcing/ peak tracking combination (red dash) predicts the dispersion behavior well, although it lies slightly above the analytical curve for the 2-3Δx waves. Neither of the IVP peak tracking curves are a good fit since they begin to fold over while the analytical behavior is monotonic. All of the LSH wave tracking/initialization combinations predict the correct trend, although the BVP initialization goes well above the analytical curve for the 2-3Δx waves. Overall, any of the LSH combinations do an adequate job of predicting the correct propagation characteristics for the SFD algorithm, and the LSH tracking method is the best choice for this algorithm.

2.4.2 Primitive finite element

Validation plots of Fourier phase errors and damping ratios and dispersion curves for the PLF algorithm are given in Figure 2.5. For the damping ratio, any of the peak

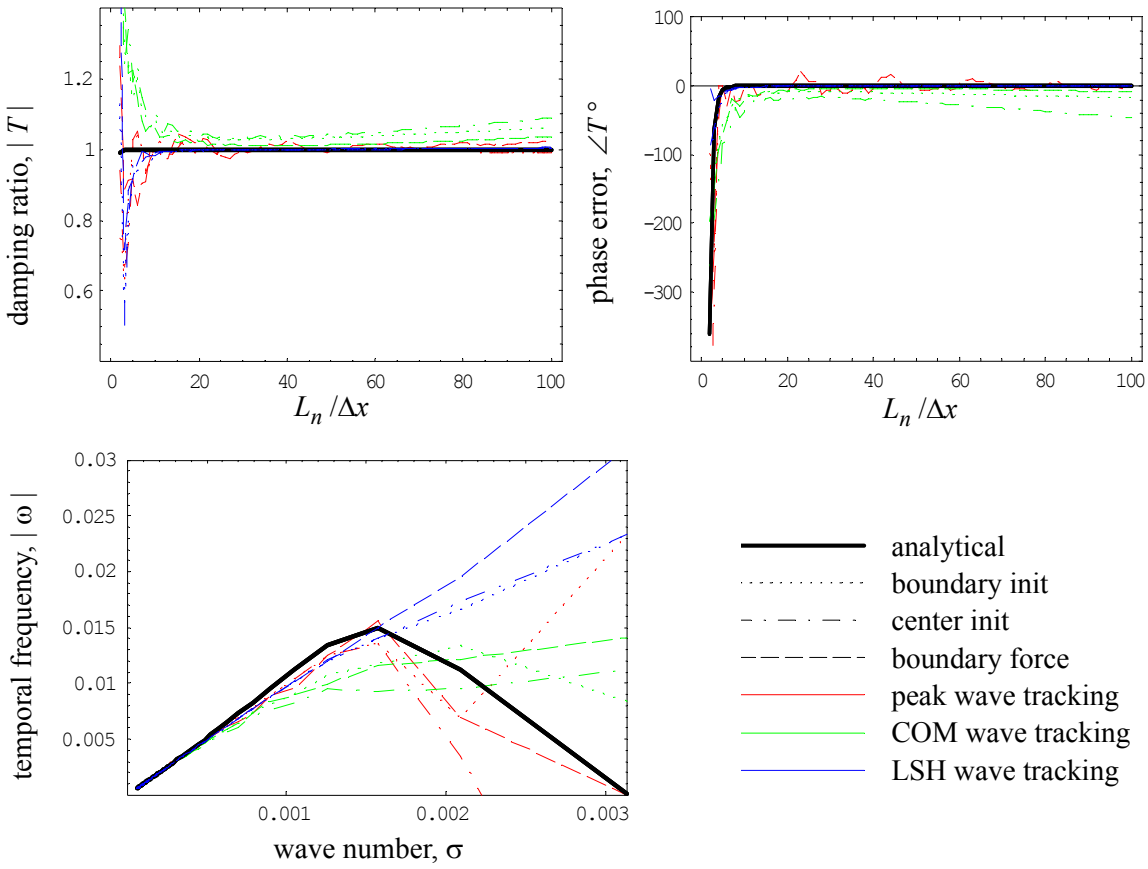


Figure 2.5 Comparison of analytical and numerical propagation properties for the PLF algorithm.

tracking/ initialization combinations match the analytical curve adequately for wavelengths greater than 10Δx, but the IVP initializations oscillate somewhat more than the BVP. None of the COM combinations match the general trend of the analytical curve, and any of the LSH are a good match for wavelengths greater than about 7Δx.

For the phase error, any of the peak tracking combinations match the analytical curve quite well, except for the BVP initialization, which oscillates about the analytical curve. The LSH wave tracking combinations match the curve well, but do not predict the correct phase lag in the 2-3Δx wavelengths. The COM tracking schemes match the general

shape of the analytical curve, but the center initialization in particular does not approach the correct asymptotic value of zero.

For the dispersion curves, only the peak tracking combinations are able to capture the folding behavior of the analytical curve, with the boundary forcing combination having the best match. Neither the COM or LSH wave tracking schemes capture the folded behavior, with the LSH scheme being the most incorrect due to the errors in the phase lags from the Fourier results. Overall, the peak tracking/ boundary forcing combination matches the dispersion best, but the LSH tracking does a better job for the Fourier characteristics.

2.4.3 ADCIRC (generalized wave continuity)

Validation plots of Fourier phase errors and damping ratios and dispersion curves for the ADCIRC model are given in Figure 2.6. For this algorithm, only the boundary forcing initialization is able to capture the correct propagation behavior for any of the characteristics. This is due to the three-time-level discretization of the GWC equation, which requires information at the past and present time levels in order to calculate the next state. Although the past wave was initialized with the same wave as the present time, but with a single time step lag, it was impossible to get a single wave to propagate with either of the IVP initialization methods; there was always two wavelengths of the wave in the domain, which was impossible for the wave tracking schemes to detect.

For the damping ratio, all three wave tracking schemes capture the analytical behavior with some error in the wavelengths smaller than $10\Delta x$. However, the LSH and peak tracking schemes best capture the asymptotic behavior. For the phase error, both the peak and LSH wave tracking schemes oscillate about the analytical curve for the small wavelengths, although the magnitude of the LSH oscillations are smaller. The COM scheme captures the behavior quite well, although it does not capture the “hook” in the $2\text{-}3\Delta x$ range. The LSH wave tracking scheme best captures the dispersion behavior, but all three tracking schemes do an adequate job for this characteristic. Overall, the boundary

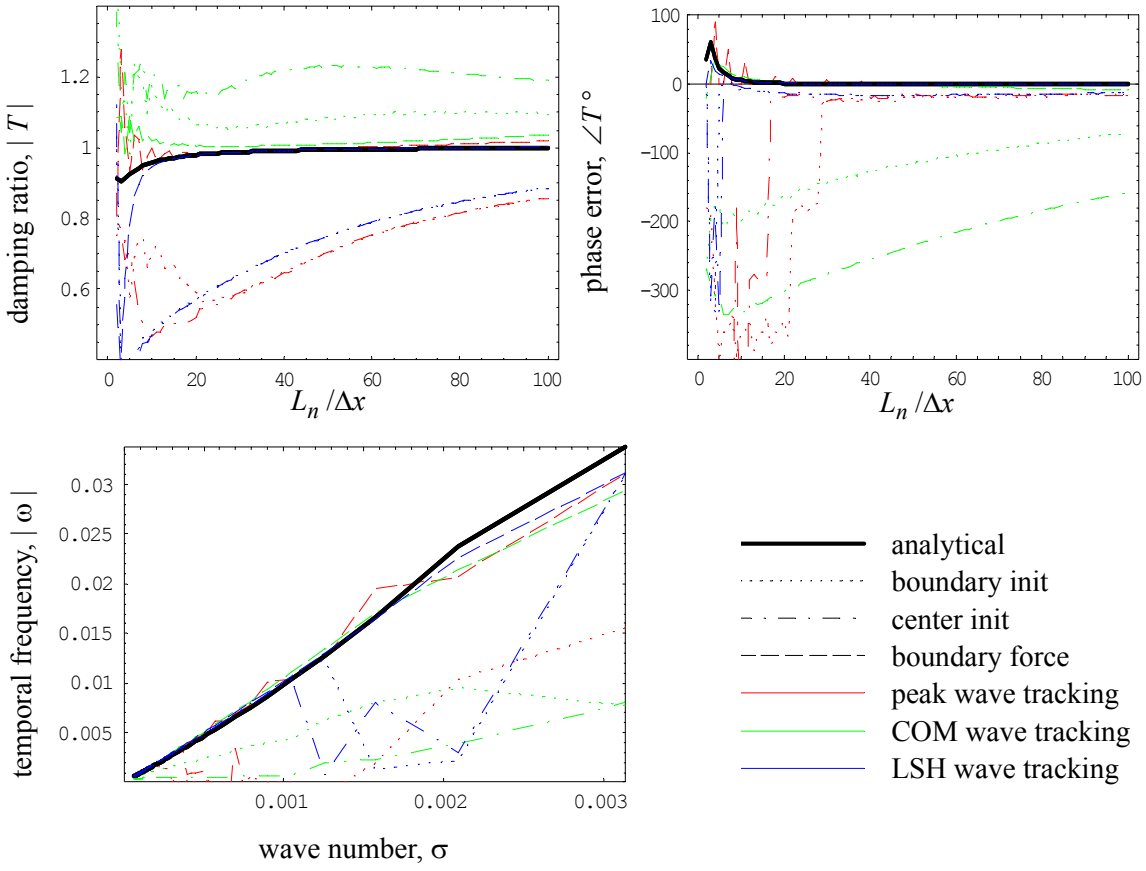


Figure 2.6 Comparison of analytical and numerical propagation properties for the ADCIRC model.

forcing initialization with LSH wave tracking does a good job of matching the analytical propagation characteristics.

2.4.4 Selective lumping finite element

Validation plots of Fourier phase errors and damping ratios for the SLFE algorithm are given in Figure 2.7. For the damping ratio, any of the initialization methods with peak or LSH wave tracking match the analytical curve for the long wavelengths. However, all of the COM wave tracking schemes predict less damping than the analytical relationship. None of the initialization/wave tracking combinations capture the analytical damping

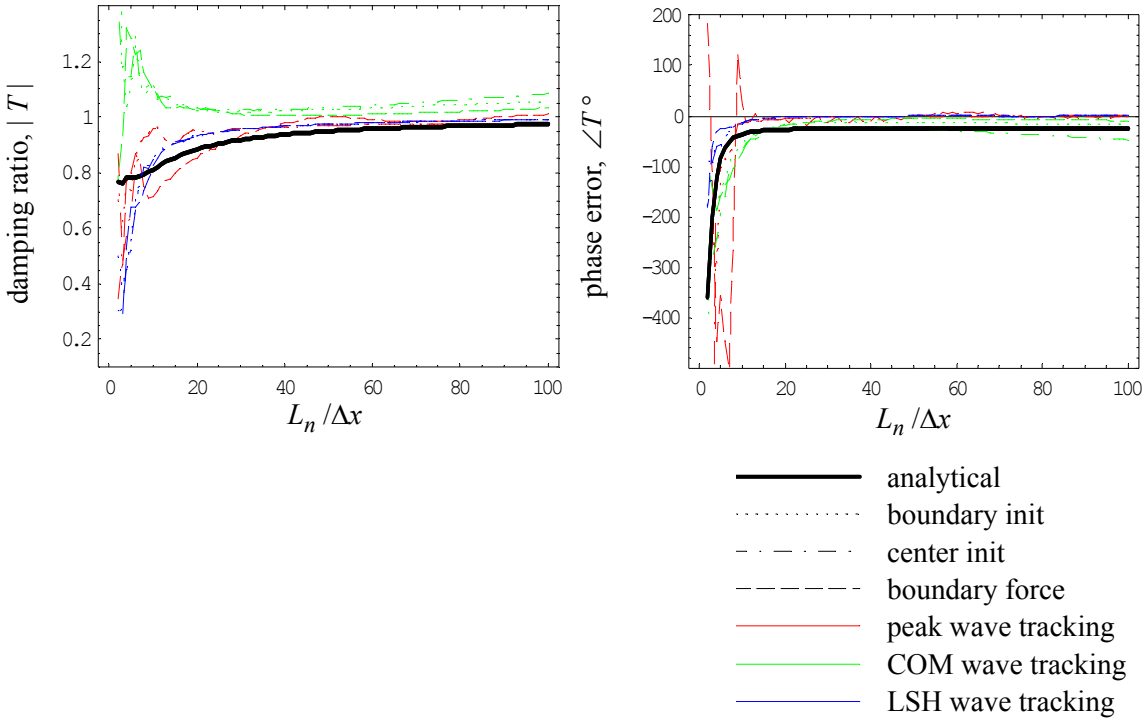


Figure 2.7 Comparison of analytical and numerical Fourier propagation properties for the SLFE algorithm.

behavior for the small wavelengths. For the phase error, none of the initialization/wave tracking combinations do a great job of matching the analytical behavior, but the COM and LSH tracking come the closest. From this phase error, it is apparent that the algorithm does better than was predicted analytically, as the predicted phase error reaches an asymptotic value of -24 degrees while the actual simulation results have an error near zero for the long, physical wavelengths.

2.4.5 Low-order finite volume

Validation plots of Fourier phase errors and damping ratios and dispersion curves for the low-order FVM algorithm are given below in Figure 2.8. Notice that the damping ratio is not well characterized by the COM wave tracking scheme, as all of the green curves

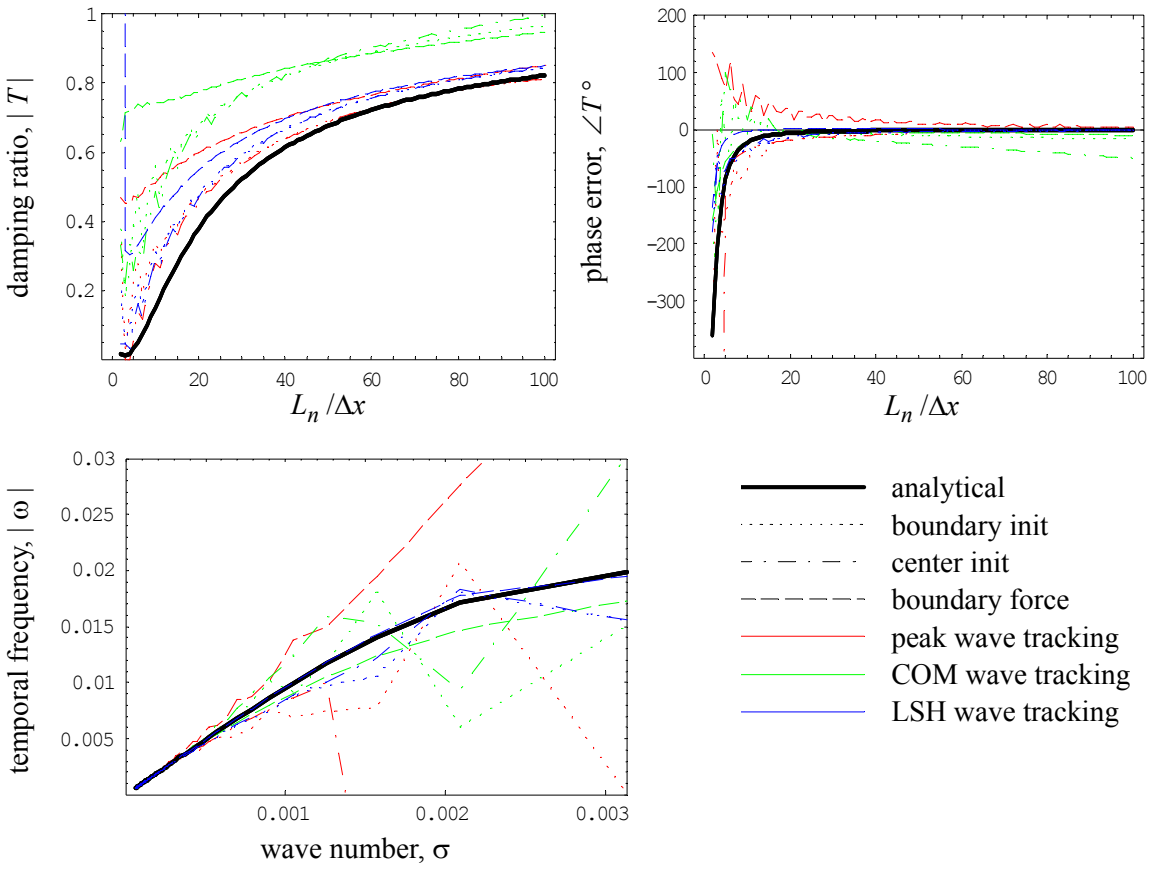


Figure 2.8 Comparison of analytical and numerical propagation properties for the low-order FVM algorithm.

lie well above the analytical curve. For peak tracking, the two IVP initialization schemes come closest to correctly predicting the damping behavior; the dotted and dash-dot red curves almost overlap each other, except in the very small wavelengths, $k < 10$. However, the boundary forcing initialization predicts less damping as the red dashed curve lies above the analytical curve. Similarly, the LSH tracking for the two IVP initializations are nearly identical, while the BVP curve lies slightly above them; however, the blue curves from the LSH wave tracking do not match the analytical damping ratio as closely as the peak tracking curves. Notice that the boundary forcing LSH curve (blue dashed) shoots up at the 2Δx wavelength; this is due to the best fit of the initial wave. Recall in Equation (2.52) that

the damping ratio is approximated by the ratio of the final amplitude over the initial amplitude to the analytical amplitude. For boundary forcing, the initial wave for the $2\Delta x$ wavelength has an amplitude close to zero so that the ratio becomes very large even though the final amplitude is also very small.

Meanwhile, the phase error is well characterized by peak tracking for either IVP but not for the boundary forcing. For the LSH wave tracking, any of the initializations adequately capture the analytical behavior, although the boundary forcing is not as good of a match. The COM wave tracking technique adequately captures the phase error when the boundary IVP or boundary forcing are used, but not when the center IVP is used.

Finally, the analytical dispersion curve is not well captured by any of the peak tracking or COM wave tracking schemes (for any initialization); but it is well captured by the LSH wave tracking, particularly for the boundary forcing initialization which almost exactly matches the analytical behavior. Notice also, that all nine of the initialization/wave tracking combinations match the analytical dispersion curve for the high wavelengths (the left-hand end of the curve) even though they do not all approach the analytical phase error of 0.0 in these wavelengths.

For the low-order FVM, peak tracking IVP methods most accurately capture the damping behavior; however, the LSH for any initialization method best captures the phase error and dispersion curves. Therefore, there is not a single combination that stands out for all three propagation characteristics.

2.4.6 Summary of qualitative validation results

The qualitative comparisons for the three initialization methods and wave tracking techniques are summarized below in Table 2.5, where the notation *good* ># indicates that the propagation results at wavelengths above this limit are almost an exact match. The overall "best" methods for each algorithm and propagation characteristic are highlighted in bold.

Table 2.5 Qualitative errors for numerical propagation initialization methods and wave tracking techniques (by SWE algorithm).

		peak tracking			center of mass			least squares harmonic		
		$\lvert T \rvert$	$\angle T$	$\lvert \omega \rvert$	$\lvert T \rvert$	$\angle T$	$\lvert \omega \rvert$	$\lvert T \rvert$	$\angle T$	$\lvert \omega \rvert$
SFD	binit	good >10	very good	good >4	bad	bad	bad	**good >10**	**very good**	**good >2**
	cinit	good	good >2	good >4	very bad	bad	bad	good >15	very good	good >2
	bforce	bad	bad	good >4	bad	bad	bad	good >10	good >2	good >3
PLF	binit	good	good >3	good >2	very bad	bad	very bad	good >5	good >3	very bad
	cinit	good	very good	good >4	very bad	bad	very bad	**good >7**	**very good**	very bad
	bforce	bad	bad	**very good**	very bad	bad	very bad	good >4	good	very bad
GWC	binit	very bad	very bad	very bad	very bad	very bad	very bad	very bad	very bad	good >4
	cinit	very bad	very bad	very bad	very bad	very bad	very bad	very bad	very bad	good >5
	bforce	good	good	good >4	good	good	good >3	**good >3**	**good**	**good >3**
SLFE *	binit	good	good	NA	bad	good	NA	good	good	NA
	cinit	good	good	NA	bad	good	NA	good	good	NA
	bforce	good	very bad	NA	bad	good	NA	good	good	NA
FVM/DG (low-order)	binit	good	good	very bad	bad	good	bad	bad	good	good
	cinit	**good**	good >4	very bad	bad	bad	bad	bad	**good >4**	good
	bforce	bad	bad	bad	bad	good	good	bad	good	**very good**

$\lvert T \rvert$ – magnitude of complex propagation factor
$\angle T$ – phase of complex propagation factor
$\lvert \omega \rvert$ – magnitude of temporal frequency
very bad – does not capture general trend of analytical results
bad – general trend is captured but does not match at any data points
good – general trend is captured and matches at many data points
very good – general trend is captured and matches at almost all data points
* There are no combinations that stand out as best for the SLFE algorithm.

From the validation plots and the summary in Table 2.5, two generalizations can be made. First, the COM wave tracking scheme is not a good indicator of the propagation properties of any of the algorithms. The predicted damping ratio is consistently higher than the analytical results, indicating that the total "mass" of the wave is not a good indicator of the amplitude. This is most likely due to the fact that the area under the curve can be increased significantly if the wave is diffused, even if the amplitude is only decreased slightly. Also, the phase error is frequently lagged relative to the analytical phase, which indicates that the center of mass is not a good indicator of wave phasing.

Secondly, the GWC algorithm does not do well with IVPs, and boundary forcing is the only way to introduce the wave such that the final wave can be analyzed. Although the boundary forcing propagation results are the most correct, of the three initialization methods, for the dispersion characteristics, they do not consistently characterize the Fourier propagation characteristics of the other SWE discretizations as well as Methods 1 and 2.

Figure 2.9 shows the initial and final waves for each algorithm and initialization method for a $75\Delta x$ wave and illustrates the difficulties with the ADCIRC model. Notice that there is a trailing wave with this model for initialization Methods 1 and 2 (cyan data points in Figures 2.9a and 2.9b). Due to the second time derivative in the derivation of the algorithm, it is difficult to satisfy the continuum equations exactly with an initial condition. In practice, a cold start is employed for this algorithm, such that the elevation and velocity fields are initialized at zero and the boundary forcing is ramped over an adequate time scale to prevent instabilities. Also in these two plots, notice that the other algorithms all have comparable phase and differ only slightly in their damping, with the low-order FVM/DG being the most dissipative, and notice that there is a slight tail on the other algorithms as well. For the boundary forcing, the PLF, SFD, SLFE and GWC curves are practically the same, while the low-order FVM/DG has a lower peak value and is more diffused.

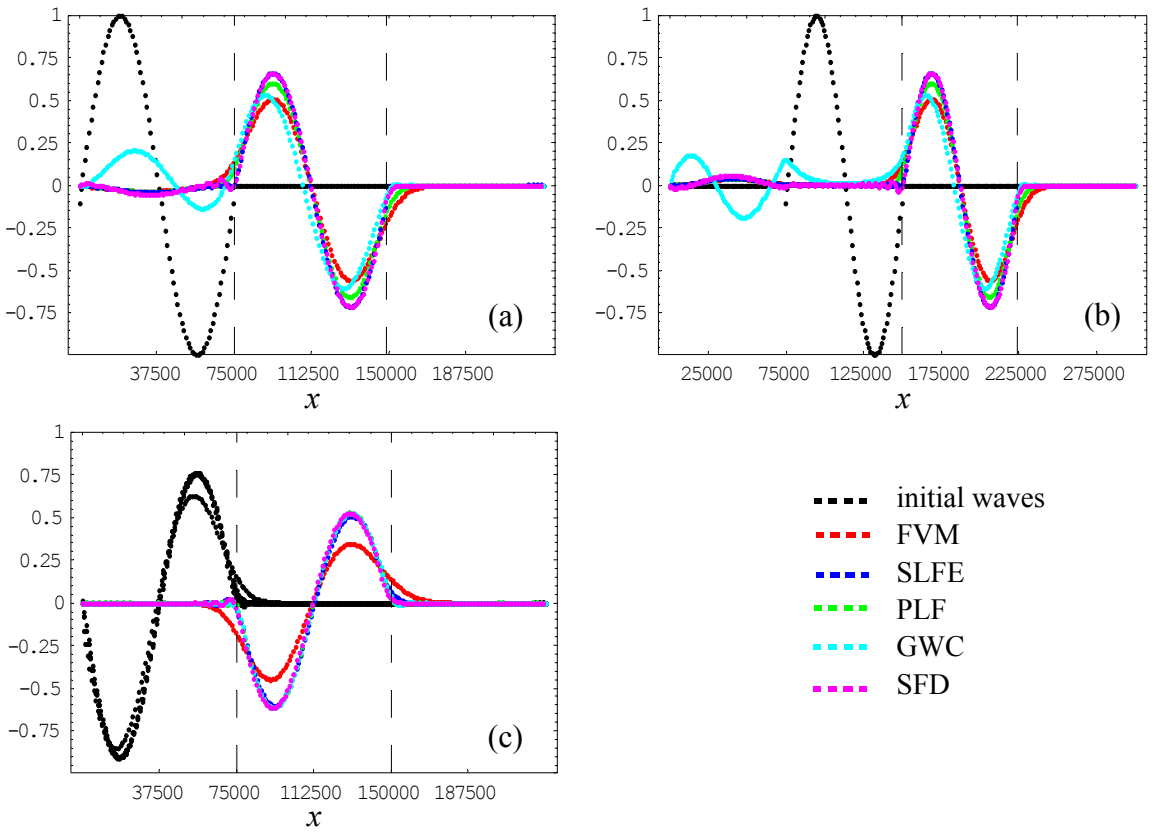


Figure 2.9 Initial and final waveforms for a 75Δx wave for all three initialization methods: (a) wave at boundary, (b) wave near center, (c) boundary forcing to bring in wave.

2.4.7 Quantitative validation of numerical propagation tools

Quantitatively, two error norms were used to characterize the error in the propagation characteristics of each algorithm. An L_2 norm for the relative error in the damping and dispersion characteristics of waves between 2 and 100 Δx was calculated as

$$\text{relative } L_2 \text{ error} = \left(\sum_{N=2}^{100} \left(\frac{true_N - approx_N}{true_N} \right)^2 / 99 \right)^{1/2} \tag{2.58}$$

where the errors are summed over the dimensionless wavelength ($N = L_n/\Delta x$), $true_N$ is the

known analytical phase characteristic for wavelength L_n, and $approx_N$ is the numerically generated phase characteristic for wavelength L_n. The relative error is a more desirable measure of error as it provides a comparison of the error in relation to the magnitude of the true characteristic. For example, if the relative error in the damping ratio for an algorithm is greater than 1.0, that means that the error is greater in magnitude than the true value itself. Conversely, if the relative error is smaller than 1.0, then the error is smaller in magnitude than the true value itself. However, the error could not be calculated in this way for the phase shift because the analytical phase shift approaches zero, which results in an undefined error measure. Therefore, an L_2 norm for the absolute error was calculated for the phase shifts as

$$\text{absolute } L_2 \text{ error} = \left(\sum_{N=2}^{100} (true_N - approx_N)^2 / 99 \right)^{1/2} \tag{2.59}$$

where the variables are the same as for the relative error.

The relative error for the damping and temporal frequency and the absolute error for the phase shift are presented below in Table 2.6 for all three initialization methods, all three wave tracking techniques and the five SWE algorithms. The relative errors in the temporal frequency for the PLF and the damping ratio for low-order FVM/DG algorithms are only computed for wavelengths N = 3 to 100, since the analytical frequency and damping ratio are near zero for the $2\Delta x$ wave. The minimum error for each propagation characteristic (damping, phase error and dispersion) and algorithm is denoted by bold italics. Note that the combinations with the minimum error may not be the same as gave the best qualitative results.

From the summary of the qualitative and quantitative error norms in Tables 2.5 and 2.6, it is evident that there is not a single initialization method or wave tracking scheme that characterizes every algorithm to within acceptable limits. However, the generalizations

Table 2.6 L_2 errors for numerical propagation initialization methods and wave tracking techniques (by SWE algorithm).

		peak tracking			center of mass			least squares harmonic		
		$\lvert T \rvert$	$\angle T$ °	$\lvert \omega \rvert$	$\lvert T \rvert$	$\angle T$ °	$\lvert \omega \rvert$	$\lvert T \rvert$	$\angle T$ °	$\lvert \omega \rvert$
SFD	binit	0.051	18.79	0.064	0.134	28.84	0.093	0.051	4.88	0.020
	cinit	0.052	23.08	0.086	0.151	64.84	0.190	***0.050***	***4.55***	***0.019***
	bforce	0.068	11.60	0.037	0.130	18.17	0.064	0.104	13.44	0.058
PLF	binit	0.046	32.66	***0.044***	0.139	19.54	0.061	***0.023***	34.29	0.054
	cinit	0.046	36.31	0.073	0.111	38.46	0.108	0.073	9.28	0.059
	bforce	0.043	19.46	0.051	0.074	22.87	0.054	0.077	***6.89***	0.078
GWC	binit	0.283	168.33	0.452	0.133	133.64	0.316	0.306	100.78	0.138
	cinit	0.316	193.75	0.518	0.222	246.08	0.652	0.310	112.03	0.146
	bforce	0.049	10.78	0.092	***0.047***	8.80	***0.084***	0.078	***7.08***	0.087
SLFE	binit	***0.058***	29.98	NA	0.249	***16.85***	NA	0.085	29.03	NA
	cinit	0.071	76.58	NA	0.198	26.47	NA	0.108	39.48	NA
	bforce	0.079	97.61	NA	0.174	34.81	NA	0.101	69.34	NA
FVM/DG (low-order)	binit	1.039	15.79	0.131	2.618	26.02	0.085	0.780	***5.21***	0.050
	cinit	***0.322***	62.76	0.436	2.727	42.71	0.122	0.517	6.77	0.042
	bforce	5.113	43.09	0.157	8.155	8.15	0.035	2.005	14.97	***0.006***

$\lvert T \rvert$ – relative L_2 error in the magnitude of complex propagation factor
$\angle T$ – absolute L_2 error in the phase of complex propagation factor
$\lvert \omega \rvert$ – relative L_2 error in the magnitude of temporal frequency

from the qualitative analysis are enforced; namely that the COM wave tracking technique generally results in greater errors for all algorithms and initialization methods, the analytical phase behavior of the SLFE algorithm is not well captured, and initialization Methods 1 and 2 result in greater errors for the GWC algorithm.

2.5 Recommendations

It has been shown that numerical analogs to traditional Fourier and dispersion analyses are able to capture the overall propagation behavior for a variety of shallow water discretization schemes. They also provide a structured framework in which to study higher-order algorithms and other schemes that do not lend themselves to traditional analysis

techniques, as will be seen in the next chapter.

Looking at both the smallest and the second smallest errors in Table 2.6, initialization Method 2 (cinit) produces reliable Fourier propagation results for all algorithms, except for the GWC algorithm. Also, initialization Method 3 produces reliable dispersion results for all algorithms. The least squares harmonic wave tracking scheme produces reliable results for all three propagation characteristics and all algorithms. However, for the primitive leap-frog algorithm, only the peak tracking scheme is able to capture the folded dispersion curve.

Therefore, when one is interested in Fourier behavior it is recommended that initialization Method 2 and least squares harmonic wave tracking be used with all algorithms except the GWC algorithm, which should use initialization Method 3 and least squares harmonic wave tracking. For dispersion behavior, initialization Method 3 is appropriate for all discretization schemes and both least squares harmonic and peak wave tracking should be examined. All results in Chapter 3 will be generated following these recommendations.

CHAPTER 3

Analysis and Comparison of Shallow Water Models in One Dimension

3.1 Introduction

Herein, a comparative study of the mathematical properties of six SWE algorithms is presented. Several analytical analysis tools exist for examining the numerical properties of linear algorithms. Of the available techniques, Fourier, dispersion, and truncation error analyses are used to examine the propagation behavior, stability and accuracy of linear algorithms. Additionally, grid convergence results are compared to the truncation errors for each of the study algorithms, and the numerical propagation tools developed in Chapter 2 are applied to the FVM, DG and SLFE algorithms, which cannot be studied analytically.

Truncation error analysis is based on a Taylor series expansion of the discrete equations; it is used to examine the accuracy and rate of convergence of numerical solutions to linear and nonlinear differential equations. This analysis is a useful tool for examining a particular solution algorithm when a closed-form discretization can be obtained for the differential equation. However, several high-order discretizations do not lead to closed-form discretizations at all times, and numerical techniques must be employed instead. One such numerical method, as outlined by *Celia and Gray* [1992], among others, is to do a grid convergence study comparing the output for various uniformly spaced grids to some known "true" solution, typically a fine grid solution utilizing small spatial and time steps. The

resulting output of each grid is compared to the true solution and some measure of error determined, often using the L_2, or other such norm. In this way, a plot of error versus grid spacing can be used to determine the order of accuracy when truncation error analysis cannot be performed analytically. Other numerical approaches include the grid convergence index of *Roache* [1998].

A good numerical solution must also remain stable over the simulation time. Stability is often examined analytically by finding the propagation factor, which is the ratio of the new value of the dependent variable to the old value at a grid point, and determining the constraints on the time step to ensure a propagation factor with magnitude less than one. The stability can also be checked numerically by running an algorithm using successively larger time steps until the solution tends toward infinity and becomes unstable. Some reasonable decimal multiple of this time step can then be used as an estimate for the stability constraint of that algorithm.

Accurate solutions also demand that the numerical algorithm does not introduce spurious modes during the solution process. The presence of such numerical noise is made evident by high frequency oscillations about the true waveform, where on average the solution is correct. Such noise limits the predictive capabilities (utility) of the model since the true solution is not easily obtainable by simple filtering. Two mathematical techniques available for examining the propagation behavior of an algorithm are Fourier analysis and dispersion analysis, which were presented in Chapter 2 and briefly reviewed here.

Fourier analysis is a useful tool to analyze the propagation behavior of solutions to homogeneous, linear difference or differential equations. If the solution is piecewise continuous on a finite interval or is periodic on an infinite interval, then it can be represented as a complex Fourier series. The key to Fourier analysis is that the linearity of the differential (or difference) equation allows one to examine a single Fourier component at a time. This type of analysis is also referred to as von Neumann analysis. This analysis

results in two related plots: phase error versus wave number, and damping ratio versus wave number. The desired behavior of a solution algorithm is that it propagates all waves in-phase, but for those waves that are out of phase, sufficient damping should be provided; too much damping results in an overly-dissipative system, while too little damping of the poorly propagated waves results in the introduction of numerical noise in the solution. Additionally, a damping ratio of magnitude greater than one indicates an unstable solution, since any noise will be amplified at each time step until the solution eventually becomes infinite. *Gray and Lynch* [1977] have used a similar approach wherein they introduce a complex propagation factor, which compares the phase and amplitude of the discrete wave to those of the analytical/continuum solution of the linear SWE.

Dispersion analysis is also based on the assumption that the solution of the differential or difference equations can be separated and expressed as periodic in time and space. However, it varies from Fourier analysis in that the time variable remains continuous by using the harmonic form of the equations. The analysis results in a dispersion curve, which is a plot of the magnitude of the temporal frequency versus wave number. The desired behavior for the SWE is a monotonic curve, which indicates a numerical solution free from spurious oscillations. A folded curve indicates aliasing of wave components, with one wave corresponding to the long physical wave while the other corresponds to short wavelength noise in the solution.

The discretizations for each SWE solution algorithm are presented in §3.2, and the complete derivations are provided in Appendices A and B. Truncation errors for the models are reported in §3.3 and confirmed with grid convergence studies in §3.4. Analytical propagation characteristics, where it is possible to derive them, are reported in §3.5, and numerical propagation results are presented for the remaining models in §3.6. Finally, a comparison of all of the models' characteristics is presented in §3.7, and some concluding remarks are given in §3.8.

3.2 Summary of discrete models for the SWE

Complete derivations for the continuous finite element models are provided in Appendix A, and those for the staggered finite difference and discontinuous models are provided in Appendix B; in the interest of brevity, only the final discrete equations are presented herein. All of the solution algorithms that will be compared in subsequent sections use the primitive continuity equation (1.5) to solve for elevations and the momentum equation (1.6) to solve for velocities, with the exception of the ADCIRC model, which uses the generalized wave continuity (GWC) equation (1.7) to solve for elevations.

3.2.1 Staggered finite difference

The discrete equations for the SFD approximation to the SWE using weighted Euler time stepping are given as

$$\zeta_j^{k+1} + h\theta\frac{\Delta t}{\Delta x}(u_{j+1/2}^{k+1} - u_{j-1/2}^{k+1}) = \zeta_j^k - h(1-\theta)\frac{\Delta t}{\Delta x}(u_{j+1/2}^k - u_{j-1/2}^k) \quad \textbf{(3.1)}$$

$$(1+\tau\Delta t\theta)u_{j+1/2}^{k+1} + g\theta\frac{\Delta t}{\Delta x}(\zeta_{j+1}^{k+1} - \zeta_j^{k+1}) =$$

$$(1-\tau\Delta t(1-\theta))u_{j+1/2}^k - g(1-\theta)\frac{\Delta t}{\Delta x}(\zeta_{j+1}^k - \zeta_j^k) \quad \textbf{(3.2)}$$

where θ is the weighting parameter and varies between 0 (fully explicit) and 1 (fully implicit). A θ value of 0.5 gives the popular Crank-Nicolson scheme.

3.2.2 Primitive finite element

The discrete equations for the PFE approximation to the SWE using leap-frog time stepping are given as

$$\zeta_{j-1}^{k+1} + 4\zeta_j^{k+1} + \zeta_{j+1}^{k+1} = (\zeta_{j-1}^{k-1} + 4\zeta_j^{k-1} + \zeta_{j+1}^{k-1}) - 6h\frac{\Delta t}{\Delta x}(u_{j+1}^k - u_{j-1}^k) \quad \textbf{(3.3)}$$

$$u_{j-1}^{k+1} + 4u_j^{k+1} + u_{j+1}^{k+1} = (1 - 2\tau\Delta t)(u_{j-1}^{k-1} + 4u_j^{k-1} + u_{j+1}^{k-1}) - 6g\frac{\Delta t}{\Delta x}(\zeta_{j+1}^{k} - \zeta_{j-1}^{k}) \quad \textbf{(3.4)}$$

where the leap-frog scheme decouples the continuity and momentum equations.

3.2.3 ADCIRC (generalized wave continuity)

The discrete equations for the linearized one-dimensional ADCIRC (ADvanced CIRCulation, *Luettich et al.* [1992, 2003]) model for the SWE, which uses a three-level temporal scheme centered at level k for continuity and a two-level temporal scheme centered at k+1/2 for momentum, are given as

$$\left(\frac{1}{\Delta t^2} + \frac{G}{2\Delta t}\right)\frac{\Delta x}{6}(\zeta_{j-1}^{k+1} + 4\zeta_j^{k+1} + \zeta_{j+1}^{k+1}) + \frac{\alpha_1 gh}{\Delta x}(-\zeta_{j-1}^{k+1} + 2\zeta_j^{k+1} - \zeta_{j+1}^{k+1}) =$$

$$\frac{2\Delta x}{6\Delta t^2}(\zeta_{j-1}^{k} + 4\zeta_j^{k} + \zeta_{j+1}^{k}) + \left(-\frac{1}{\Delta t^2} + \frac{G}{2\Delta t}\right)\frac{\Delta x}{6}(\zeta_{j-1}^{k-1} + 4\zeta_j^{k-1} + \zeta_{j+1}^{k-1}) -$$

$$(G - \tau)\frac{h}{2}(u_{j+1}^{k} - u_{j-1}^{k}) - \frac{\alpha_2 gh}{\Delta x}(-\zeta_{j-1}^{k} + 2\zeta_j^{k} - \zeta_{j+1}^{k}) -$$

$$\frac{\alpha_3 gh}{\Delta x}(-\zeta_{j-1}^{k-1} + 2\zeta_j^{k-1} - \zeta_{j+1}^{k-1}) + B^k \quad \textbf{(3.5)}$$

$$\Delta x\left(1 + \frac{\tau\Delta t}{2}\right)u_j^{k+1} = \Delta x\left(1 - \frac{\tau\Delta t}{2}\right)u_j^{k} - \frac{g\Delta t}{4}(\zeta_{j+1}^{k+1} + \zeta_{j+1}^{k} - (\zeta_{j-1}^{k+1} + \zeta_{j-1}^{k})) \quad \textbf{(3.6)}$$

where the temporal weighting parameters, α_n, must satisfy $\alpha_1+\alpha_2+\alpha_3 = 1.0$, and they are usually set equal to 1/3. Rather than solving both equations simultaneously, the GWC equation is solved for the new elevations, which are then used to calculate the new velocities using the momentum equation.

3.2.4 Selective lumping finite element

The discrete equations for the SLFE approximation to the SWE, which uses a two-step temporal discretization, are given as

$$\zeta_j^{k+1} = \left[\frac{2+e}{3} - \frac{\mu^2 gh}{4}\right]\zeta_j^k + \frac{1-e}{6}(\zeta_{j-1}^k + \zeta_{j+1}^k) + \frac{\mu^2 gh}{8}(\zeta_{j-2}^k + \zeta_{j+2}^k) -$$

$$\frac{\mu h}{2}\left[\frac{1-e}{6} - \frac{\tau\Delta t}{2}\frac{1-f}{6}\right](u_{j+2}^k - u_{j-2}^k) - \frac{\mu h}{2}\left[\frac{2+e}{3} - \frac{\tau\Delta t}{2}\frac{2+f}{3}\right](u_{j+1}^k - u_{j-1}^k) \quad \textbf{(3.7)}$$

$$u_j^{k+1} = \left[\frac{\mu^2 gh}{8} - \tau\Delta t\frac{1-e}{6}\frac{1-f}{6} + \frac{\tau^2\Delta t^2}{2}\frac{(1-f)^2}{36}\right](u_{j-2}^k + u_{j+2}^k) +$$

$$\left[\frac{1-e}{6} - \tau\Delta t\left(\frac{1-f}{6}\frac{2+e}{3} + \frac{2+f}{3}\frac{1-e}{6} - \tau\Delta t\frac{1-f}{6}\frac{2+f}{3}\right)\right](u_{j-1}^k + u_{j+1}^k) +$$

$$\left[\frac{2+e}{3} - \frac{\mu^2 gh}{4} + 2\tau\Delta t\frac{1-f}{6}\left(\frac{\tau\Delta t}{2}\frac{1-f}{6} - \frac{1-e}{6}\right) + \tau\Delta t\frac{2+f}{3}\left(\frac{\tau\Delta t}{2}\frac{2+f}{3} - \frac{2+e}{3}\right)\right]u_j^k -$$

$$\frac{\mu g}{2}\left[\frac{1-e}{6} - \frac{\tau\Delta t}{2}\frac{1-f}{6}\right](\zeta_{j+2}^k - \zeta_{j-2}^k) - \frac{\mu g}{2}\left[\frac{2+e}{3} - \frac{\tau\Delta t}{2}\frac{2+f}{3}\right](\zeta_{j+1}^k - \zeta_{j-1}^k) \quad \textbf{(3.8)}$$

where e is the general selective lumping parameter, which varies between fully lumped ($e = 1$) and unlumped ($e = 0$), and f is the selective lumping parameter for the bottom friction term, which takes on the same range of values as e. These equations were derived by substituting the first step into the second step in order to get a single equation for analysis purposes. See Appendix A for the equations for each individual step.

3.2.5 Finite volume

The general discrete equations for the FVM approximation to the SWE using explicit time stepping are given as

$$\zeta_j^{k+1} = \zeta_j^k - \frac{\Delta t}{2\Delta x}[h(u_{j,R}^k + u_{j+1,L}^k - u_{j-1,R}^k - u_{j,L}^k) -$$

$$a(\zeta_{j-1,R}^k - \zeta_{j,L}^k - \zeta_{j,R}^k + \zeta_{j+1,L}^k)] \quad \textbf{(3.9)}$$

$$u_j^{k+1} = u_j^k - \frac{\Delta t}{2\Delta x}[g(\zeta_{j,R}^k + \zeta_{j+1,L}^k - \zeta_{j-1,R}^k - \zeta_{j,L}^k)$$

$$-a(u^k_{j-1,R} - u^k_{j,L} - u^k_{j,R} + u^k_{j+1,L})] - \tau u^k_j \Delta t\,, \tag{3.10}$$

where the subscripts $_{j,\,R(L)}$ indicate the rightmost (leftmost) value of u or ζ within volume j and a is the frictionless wave celerity and is equal to $\sqrt{gh}$. These equations are valid for any order of approximation, as the order does not enter in until the left and right states are calculated.

Looking at Equations (3.9) and (3.10), and noting that for piecewise constants $u_{j,L} = u_{j,R}$ and $\zeta_{j,\,L} = \zeta_{j,\,R}$, the low-order discrete FVM representation can be written as

$$\zeta^{k+1}_j = \zeta^k_j - \frac{\Delta t}{2\Delta x}[h(u^k_{j+1} - u^k_{j-1}) - a(\zeta^k_{j-1} - 2\zeta^k_j + \zeta^k_{j+1})] \tag{3.11}$$

$$u^{k+1}_j = (1 - \tau\Delta t)u^k_j - \frac{\Delta t}{2\Delta x}[g(\zeta^k_{j+1} - \zeta^k_{j-1}) - a(u^k_{j-1} - 2u^k_j + u^k_{j+1})]\,. \tag{3.12}$$

For the higher-order FVM with piecewise linear approximations, some limiting procedure is necessary to maintain stability. In this study, three common limiters (as presented in various applications of higher-order methods) and their behavior within the context of the high-order FVM discretization of the SWE have been examined. These are the minmod limiter used by *Sweby* [1984] given as

$$\text{minmod slope} = s_L \text{Min}(|L|, |R|) \quad \text{if } s_L s_R > 0, \quad \text{else slope} = 0\,, \tag{3.13}$$

the Superbee limiter used by *Causon et al.* [2000] given as

$$\text{Superbee slope} = s_L \text{Max}[0, \text{Min}(2|L|, s_L|R|), \text{Min}(|L|, 2s_L|R|)]\,, \tag{3.14}$$

and the vanLeer limiter used by *Bell and Shubin* [1984] given as

$$\text{vanLeer slope} = 0.5 s_C \text{Min}(0.5C, \text{Minmod}(L, R))\,, \tag{3.15}$$

where s_x is the sign of x and the raw slopes are calculated using standard finite differences:

$$L = \frac{b_j - b_{j-1}}{\Delta x},\ R = \frac{b_{j+1} - b_j}{\Delta x},\ \text{and}\ C = \frac{b_{j+1} - b_{j-1}}{2\Delta x} \tag{3.16}$$

where b can be either u or ζ, the subscripts indicate the cell average values for the j^{th} cell, and constant grid spacing Δx is assumed. See Appendix B for more details and a discussion of slope limiting for the higher-order FVM.

3.2.6 Discontinuous Galerkin

The discrete equations for the low-order DG approximation to the SWE using explicit time stepping are given as

$$\zeta_j^{k+1} = \zeta_j^k + \frac{\Delta t}{2\Delta x}[h(u_{j-1}^k + 2u_j^k + u_{j+1}^k) - a(\zeta_{j+1}^k - \zeta_{j-1}^k)] \tag{3.17}$$

$$u_j^{k+1} = (1 - \tau\Delta t)u_j^k + \frac{\Delta t}{2\Delta x}[g(\zeta_{j-1}^j + 2\zeta_j^k + \zeta_{j+1}^k) - a(u_{j+1}^k - u_{j-1}^k)]\ , \tag{3.18}$$

which are the same as the low-order FVM.

The discrete equations for the higher-order (piecewise linear) DG approximation to the SWE using explicit time stepping are given as

$$\begin{aligned}
\zeta_j^{(0)k+1} &= \zeta_j^{(0)k} + \frac{\Delta t}{2\Delta x_j}[h(u_{j-1}^{(0)k} + 6u_{j-1}^{(1)k} + u_j^{(0)k} - 6u_j^{(1)k}) - \\
&a(\zeta_j^{(0)k} - 6\zeta_j^{(1)k} - (\zeta_{j-1}^{(0)k} + 6\zeta_{j-1}^{(1)k}))] - \frac{\Delta t}{2\Delta x_j}[h(u_j^{(0)k} + 6u_j^{(1)k} + u_{j+1}^{(0)k} - 6u_{j+1}^{(1)k}) - \\
&a(\zeta_{j+1}^{(0)k} - 6\zeta_{j+1}^{(1)k} - (\zeta_j^{(0)k} + 6\zeta_j^{(1)k}))]
\end{aligned} \tag{3.19}$$

$$\begin{aligned}
u_j^{(0)k+1} &= (1 - \tau\Delta t)u_j^{(0)k} + \frac{\Delta t}{2\Delta x_j}[g(\zeta_{j-1}^{(0)k} + 6\zeta_{j-1}^{(1)k} + \zeta_j^{(0)k} - 6\zeta_j^{(1)k}) - \\
&a(u_j^{(0)k} - 6u_j^{(1)k} - (u_{j-1}^{(0)k} + 6u_{j-1}^{(1)k}))] - \frac{\Delta t}{2\Delta x_j}[g(\zeta_j^{(0)k} + 6\zeta_j^{(1)k} + \zeta_{j+1}^{(0)k} - 6\zeta_{j+1}^{(1)k}) -
\end{aligned}$$

$$a(u_{j+1}^{(0)k} - 6u_{j+1}^{(1)k} - (u_j^{(0)k} + 6u_j^{(1)k}))] \tag{3.20}$$

$$\zeta_j^{(1)k+1} = \zeta_j^{(1)k} + \frac{\Delta t}{\Delta x_j} h u_j^{(0)k} - \frac{\Delta t}{4\Delta x_j}[h(u_{j-1}^{(0)k} + 6u_{j-1}^{(1)k} + u_j^{(0)k} - 6u_j^{(1)k}) -$$

$$a(\zeta_j^{(0)k} - 6\zeta_j^{(1)k} - (\zeta_{j-1}^{(0)k} + 6\zeta_{j-1}^{(1)k}))] - \frac{\Delta t}{4\Delta x_j}[h(u_j^{(0)k} + 6u_j^{(1)k} + u_{j+1}^{(0)k} - 6u_{j+1}^{(1)k}) -$$

$$a(\zeta_{j+1}^{(0)k} - 6\zeta_{j+1}^{(1)k} - (\zeta_j^{(0)k} + 6\zeta_j^{(1)k}))] \tag{3.21}$$

$$u_j^{(1)k+1} = (1 - \tau\Delta t)u_j^{(1)k} + \frac{\Delta t}{\Delta x_j} g\zeta_j^{(0)k} - \frac{\Delta t}{4\Delta x_j}[g(\zeta_{j-1}^{(0)k} + 6\zeta_{j-1}^{(1)k} + \zeta_j^{(0)k} - 6\zeta_j^{(1)k}) -$$

$$a(u_j^{(0)k} - 6u_j^{(1)k} - (u_{j-1}^{(0)k} + 6u_{j-1}^{(1)k}))] - \frac{\Delta t}{4\Delta x_j}[g(\zeta_j^{(0)k} + 6\zeta_j^{(1)k} + \zeta_{j+1}^{(0)k} - 6\zeta_{j+1}^{(1)k}) -$$

$$a(u_{j+1}^{(0)k} - 6u_{j+1}^{(1)k} - (u_j^{(0)k} + 6u_j^{(1)k}))] \ , \tag{3.22}$$

where the parenthetical superscripts (0) and (1) represent the two degrees of freedom (DOF) for each state variable. The zero DOF is the element average for the variable and the one DOF is the slope within the element. The k superscripts are from the temporal discretization and indicate at which time level the terms should be evaluated.

As in the higher-order FVM, slope limiters may be necessary for added stability; however, of the limiters that were used in the FVM, only the minmod limiter will be applied to the higher-order DG in this study. (The minmod limiter with a total variation bounded (TVB) correction was used in the original development of the Runge-Kutta DG method by *Cockburn and Shu* [1991].) Additionally, some changes are necessary relative to the limiting procedure used for the FVM: 1) the limiter must compare the calculated DG slopes to the left and right slopes and 2) the centered slopes C are taken to be the DG state values times the normalizing factors, $(12/\Delta x_j)\zeta_j^{(1)}$ and $(12/\Delta x_j)u_j^{(1)}$, rather than calculating them as a centered difference. With these changes, the minmod limiter in the DG framework is given as

$$\text{minmod slope} = s_C \text{Min}(|L|, |C|, |R|) \quad \text{if } s_L = s_C = s_R \text{ , else slope} = 0 \tag{3.23}$$

where the left and right slopes are calculated using finite differences and element averages from adjacent elements, as in the FVM. For non-constant grid spacing, the limiter is expressed as given in Equation (3.23); but the slopes are calculated as

$$L = \frac{b_j^{(0)} - b_{j-1}^{(0)}}{0.5(\Delta x_j + \Delta x_{j-1})}, \; C = \frac{12}{\Delta x_j} b_j^{(1)}, \text{ and } R = \frac{b_{j+1}^{(0)} - b_j^{(0)}}{0.5(\Delta x_{j+1} + \Delta x_j)} \tag{3.24}$$

where b can be either state variable, ζ or u. After the slopes are limited, the new value for the (1) degree of freedom is given as

$$b_j^{(1)} = \frac{\Delta x_j}{12} \text{slope}. \tag{3.25}$$

See Appendix B for the complete derivation of the DG algorithm and a discussion of slope limiting.

3.3 Truncation error analysis

Taylor series expansions of the discrete equations presented in §3.2 can be used to determine the order of the spatial and temporal accuracy for each algorithm that can be written in a closed form. These expansions can be thought of as the sum of the continuum equation plus some truncation error, *TE*. Thus the error for either the continuity or the momentum approximation can be written as

$$TE = error = continuum - approximation, \tag{3.26}$$

where for a consistent approximation, the *TE* must approach zero as Δt and Δx approach zero. Also, the order of the leading spatial and temporal terms of the *TE* denote the accuracy of the approximation.

In the following subsections, the truncation error will be presented for each of the

algorithms and the accuracy determined. These errors were derived by substituting a Taylor series approximation in space and time about point $x^* = (x_j, t^k)$ given as

$$b(x,t) = b(x_j,t^k) + (x-x_j)\left.\frac{\partial b}{\partial x}\right|_{x^*} + (t-t^k)\left.\frac{\partial b}{\partial t}\right|_{x^*} + \left.\frac{(x-x_j)^2}{2!}\frac{\partial^2 b}{\partial x^2}\right|_{x^*} +$$

$$(x-x_j)(t-t^k)\left.\frac{\partial^2 b}{\partial x \partial t}\right|_{x^*} + \left.\frac{(t-t^k)^2}{2!}\frac{\partial^2 b}{\partial t^2}\right|_{x^*} + \frac{(x-x_j)^{N-m}}{(N-m)!}\frac{(t-t^k)^m}{m!}\frac{\partial^N b}{\partial x^{N-m}\partial t^m} + R^{N+1} \quad \textbf{(3.27)}$$

where $b(x, t)$ can be either velocity or elevation and can be evaluated at any location or time, N is the order of the approximation, and R^{N+1} is a multidimensional remainder term that is often written as *HOT* for higher-order terms. The continuum equations are subtracted from these approximations and the truncation error simplified to give the final results presented below. In order to save space, subscript notation is used to indicate partial derivatives, and it is understood that all of the elevation and velocity derivatives are evaluated at x^*. The accuracy of these approximations, as found in this analysis, are also summarized in the final algorithm comparison given in Table 3.9. A third order, $N = 3$, Taylor series approximation for all terms in the discretizations was sufficient to determine the spatial and temporal accuracy for all of the algorithms.

3.3.1 Staggered finite difference

The truncation error for the continuity equation, as discretized by the staggered finite difference algorithm, is

$$TE = -\frac{\Delta t}{2}\zeta_{tt} - \frac{\Delta t^2}{6}\zeta_{ttt} - \Delta t h\theta u_{xt} - \frac{\Delta t^2}{2}h\theta u_{xtt} - \frac{\Delta x^2}{24}h u_{xxx} + HOT\ , \quad \textbf{(3.28)}$$

which can be further simplified by using the continuity equation and noting that $L_t \equiv \zeta_{tt} + h u_{xt} = 0$. Thus, for Crank-Nicolson time weighting with $\theta = 1/2$, one can eliminate the terms, $\frac{\Delta t}{2}\zeta_{tt} + \frac{\Delta t}{2}h u_{xt} = \frac{\Delta t}{2}L_t \equiv 0$, to get

$$TE = -\frac{\Delta t^2}{6}\zeta_{ttt} - \frac{\Delta t^2}{4}hu_{xtt} - \frac{\Delta x^2}{24}hu_{xxx} + HOT \ , \tag{3.29}$$

which is second-order accurate in both time and space. The truncation error for the momentum equation discretization for this algorithm is

$$TE = -\Delta t\theta\tau u_t - \frac{\Delta t}{2}(1 + \Delta t\theta\tau)u_{tt} - \Delta t\theta g\zeta_{xt} - \frac{\Delta t^2}{6}(1 + \Delta t\theta\tau)u_{ttt} -$$

$$\frac{\Delta t^2}{2}\theta g\zeta_{xtt} - \frac{\Delta x^2}{24}g\zeta_{xxx} + HOT \ , \tag{3.30}$$

where the Taylor approximations were centered about j =1/2 since that is the node of interest for the velocity solution. As in the continuity approximation, this expression can be simplified by using the momentum equation and noting that $M_t \equiv u_{tt} + \tau u_t + g\zeta_{xt} = 0$. Thus, for Crank-Nicolson time weighting with θ = 1/2, one can eliminate the terms, $\frac{\Delta t}{2}u_{tt} + \frac{\Delta t}{2}\tau u_t + \frac{\Delta t}{2}g\zeta_{xt} = \frac{\Delta t}{2}M_t \equiv 0$, to get

$$TE = -\frac{\Delta t^2}{4}\tau u_{tt} - \frac{\Delta t^2}{6}\left(1 + \frac{\Delta t}{2}\tau\right)u_{ttt} - \frac{\Delta t^2}{4}g\zeta_{xtt} - \frac{\Delta x^2}{24}g\zeta_{xxx} + HOT \ , \tag{3.31}$$

which is second-order accurate in both time and space.

3.3.2 Primitive finite element

The truncation error for the continuity equation, as discretized by the primitive finite element algorithm with leap-frog time stepping, is

$$TE = -\frac{\Delta x^2}{6}hu_{xxx} - \frac{\Delta x^2}{6}\zeta_{xxt} - \frac{\Delta t^2}{6}\zeta_{ttt} + HOT \ , \tag{3.32}$$

which is second-order accurate in both time and space. The truncation error for the momentum equation discretization for this algorithm is

$$TE = \Delta t\tau u_t - \frac{\Delta x^2}{6}\tau u_{xx} - \frac{\Delta t^2}{2}\tau u_{tt} - \frac{\Delta x^2}{6}(1-\tau\Delta t)u_{xxt} -$$

$$\frac{\Delta x^2}{6} g\zeta_{xxx} - \frac{\Delta t^2}{6}(1-\tau\Delta t)u_{ttt} + HOT\ , \qquad \textbf{(3.33)}$$

which is first-order accurate in time and second-order accurate in space.

3.3.3 ADCIRC (generalized wave continuity)

The truncation error for the generalized wave continuity equation, as discretized by the ADCIRC model with $\alpha_1 = \alpha_2 = \alpha_3 = 1/3$, is

$$TE = -\frac{\Delta x^2}{6} G\zeta_{xxt} - \frac{\Delta x^2}{6} h(G-\tau)u_{xxx} - \frac{\Delta t^2}{6} G\zeta_{ttt} + HOT \qquad \textbf{(3.34)}$$

which is second-order accurate in both time and space. This combination of time weighting parameters gives the optimal accuracy. The truncation error for the momentum equation, as discretized by this model, is

$$TE = -\frac{\Delta t}{2}\tau u_t - \frac{\Delta t}{2} g\zeta_{xt} - \frac{\Delta t}{2}\left(1+\frac{\tau\Delta t}{2}\right)u_{tt} - \frac{\Delta t^2}{4} g\zeta_{xtt} -$$

$$\frac{\Delta x^2}{6} g\zeta_{xxx} - \frac{\Delta t^2}{6}\left(1+\frac{\tau\Delta t}{2}\right)u_{ttt} + HOT\ , \qquad \textbf{(3.35)}$$

which can be further simplified by using the momentum equation and noting that $M_t \equiv u_{tt} + \tau u_t + g\zeta_{xt} = 0$. Thus the terms, $\frac{\Delta t}{2}\tau u_t + \frac{\Delta t}{2} g\zeta_{xt} + \frac{\Delta t}{2} u_{tt} = \frac{\Delta t}{2} M_t \equiv 0$, can be eliminated to get

$$TE = -\frac{\Delta t^2}{4}\tau u_{tt} - \frac{\Delta t^2}{4} g\zeta_{xtt} - \frac{\Delta x^2}{6} g\zeta_{xxx} - \frac{\Delta t^2}{6}\left(1+\frac{\tau\Delta t}{2}\right)u_{ttt} + HOT\ , \qquad \textbf{(3.36)}$$

which is second-order accurate in both time and space.

3.3.4 Selective lumping finite element

The truncation error for the continuity equation, as discretized by the selective lumping finite element algorithm (when the first step is substituted into the second step to get a single equation), is

$$TE = \frac{\Delta t}{2} h\tau u_x + \left(\frac{1-e}{6}\frac{\Delta x^2}{\Delta t} + \frac{\Delta t}{2} gh\right)\zeta_{xx} - \frac{\Delta t}{2}\zeta_{tt} -$$

$$\frac{\Delta x^2}{6} h\left[2 - e - \tau\Delta t\left(1 - \frac{f}{2}\right)\right]u_{xxx} - \frac{\Delta t^2}{6}\zeta_{ttt} + HOT\ . \qquad \textbf{(3.37)}$$

This approximation is only consistent with the continuum equation when $e = 1$ and all of the mass matrices in the continuity approximation are fully lumped. Under these conditions, it is first-order accurate in time and second-order accurate in space.

The truncation error for the momentum equation discretization for this algorithm is

$$TE = -\left[\frac{\Delta x^2}{6}\left(\tau(1-f) - \frac{1-e}{\Delta t}\right) - \frac{\Delta t}{2} gh\right]u_{xx} - \frac{\Delta t}{2} u_{tt} -$$

$$\frac{\Delta x^2}{6} g(2-e)\zeta_{xxx} - \frac{\Delta t^2}{6} u_{ttt} + HOT, \qquad \textbf{(3.38)}$$

Again the momentum discretization is only consistent when the mass matrices are fully lumped in both steps, $e = 1$. Under these conditions, it is first-order accurate in time and second-order accurate in space.

Kawahara et al. [1982] note that in practice a value between 0.8 and 0.95 is used for the lumping parameter, and the maximum stable time step must be used to reduce the numerical damping. However, in light of these TE results, the large time step and lumping parameter values are necessary to reduce the importance of the inconsistent $(1 - e)\Delta x^2/\Delta t$ terms in the error expression. This will be discussed further when the spatial and temporal convergence results are presented for this algorithm in §3.4. Additionally, the effect of the

lumping parameter and time step on the Fourier propagation behavior of the SLFE algorithm are examined in §3.5.2.

3.3.5 Low-order finite volume / discontinuous Galerkin

The truncation error for the continuity equation, as discretized by the low-order finite volume or low-order discontinuous Galerkin algorithm (they are equivalent) with explicit time-stepping, is

$$TE = \frac{\Delta x}{2} a\zeta_{xx} - \frac{\Delta t}{2}\zeta_{tt} - \frac{\Delta x^2}{6} h u_{xxx} - \frac{\Delta t^2}{6}\zeta_{ttt} + HOT \ , \tag{3.39}$$

which is first-order accurate in time and space. The truncation error for the momentum equation discretization for these algorithms is

$$TE = \frac{\Delta x}{2} a u_{xx} - \frac{\Delta t}{2} u_{tt} - \frac{\Delta x^2}{6} g\zeta_{xxx} - \frac{\Delta t^2}{6} u_{ttt} + HOT \ , \tag{3.40}$$

which is also first-order accurate in time and space.

It is instructive to compare the discrete FVM equations with their continuum counterparts. Equations (3.11) and (3.12) for the low-order FVM are repeated here for convenience

$$\zeta_j^{k+1} = \zeta_j^k - \frac{\Delta t}{2\Delta x}[h(u_{j+1}^k - u_{j-1}^k) - a(\zeta_{j-1}^k - 2\zeta_j^k + \zeta_{j+1}^k)] \tag{3.41}$$

$$u_j^{k+1} = (1 - \tau\Delta t)u_j^k - \frac{\Delta t}{2\Delta x}[g(\zeta_{j+1}^k - \zeta_{j-1}^k) - a(u_{j-1}^k - 2u_j^k + u_{j+1}^k)] \, . \tag{3.42}$$

Note that the low-order FVM is basically a central difference approximation that introduces a second-order space derivative into the discrete equations, viz. the terms of Equations (3.41) and (3.42) that are preceded by the parameter *a*. *Kinnmark* [1986] and *Atkinson et al.* [2004] have noted that a characteristic feature of the generalized wave continuity and

quasi-bubble algorithms, which produce non-oscillatory solutions, is the presence of a second-order space derivative. This derivative is correlated with the ability of an algorithm to successfully propagate or damp out undesirable short wavelength oscillations. *Chippada et al.* [1998] note that a centered spatial approximation to the first derivative, which is derived by using an arithmetic average for the flux approximation, results in an unstable algorithm, i.e., neglecting the terms preceded by a in Equations (3.41) and (3.42). Thus the second-order terms generated by the solution to the Riemann problem stabilize the algorithm.

Unlike the GWC and quasi-bubble algorithms, which manipulate the governing equations in continuum (GWC) or discrete (quasi-bubble) forms, the Taylor series analysis on Equations (3.41) and (3.42) reveals that the Godunov type flux approximation, using Roe's linearization, introduces the second-order spatial derivative in the leading term of the truncation error. In other words, the discrete equations are still consistent with the original continuum equations, since the second-order spatial derivative is introduced to the leading term of the error instead of in the equations themselves. This additional term has the effect of reducing the overall accuracy of the approximation to first-order, and it adds diffusion to both the continuity and momentum equations. It also stabilizes the algorithm despite the fact that the first-order spatial derivatives come from the arithmetic average of the left and right states. The overly-dissipative nature of this algorithm will be shown in §3.6.

3.4 Numerical error analysis

For both the spatial and temporal convergence studies, the test case was a 1D channel with M_2 tidal forcing (period = 44712 sec) of amplitude equal to 1m on the ocean boundary and a land boundary on the other end. The channel dimensions are 50km by 10m. The initial condition was a cold start, where the elevation and velocity fields are zero at time t=0; and the simulations were run out for slightly more than two full periods of the tidal forcing for a total of 90,000 seconds. Since the initial condition for the domain is a

cold start for both the elevation and velocity fields, only the last time output is used to compute the errors. This allows the entire domain to spin-up before examining the output.

3.4.1 Spatial accuracy

For the spatial convergence studies, a fine grid solution (1281 nodes) with a time step of 0.01 seconds from the ADCIRC model was used for the "true" solution. In order to isolate the spatial errors, a time step of 0.01 seconds was used for all of the other simulations as well, except for the SLFE, which requires the maximum stable step for optimal (consistent) results.

Two error norms were computed for the final output at t_f= 90,000 seconds: 1) an L_2 error norm in space, computed as

$$L_2 \text{ error } = \left(\left(\sum_{j=1}^{nn} (B(x_j, t_f) - b_j(t_f))^2 \right) / nn \right)^{1/2} ; \quad \textbf{(3.43)}$$

and 2) an L_∞ error norm in space, computed as

$$L_\infty \text{ error } = max[(B(x_j, t_f) - b_j(t_f))^2]^{1/2}, \quad \textbf{(3.44)}$$

where nn is the number of nodes in the approximation, b (B) can be either elevation or velocity, the $b_j(t_f)$ are the simulation results at node j and time t_f=90,000 seconds, and the true solutions $B(x, t)$ are interpolated functions from the fine-grid ADCIRC solution. When these error norms are plotted against the spatial resolution in log-log space, the slope represents the accuracy of the algorithm.

The "slopes" are computed for each of the error norms in three ways: 1) a best-fit line through the linear portion of the log-log convergence plots, 2) the average of the slopes computed between successive grid resolutions, and 3) the peak (or maximum) of the slopes computed between successive grid resolutions. Thus, 6 approximations to the spatial

accuracy are computed for each of the state variables, elevation and velocity.

A graphic presentation of the results provides for easier comparison between algorithms, therefore only the log-log plots will be presented in this section. Each of the figures will present both of the error norms for the elevation and velocity results, for a total of 4 curves per plot. The tabular data used to generate these plots, as well as the peak, average, and best-fit linear convergence rates, are available to the interested reader in Appendix D. Also, the spatial accuracy of each algorithm, as computed from the best-fit and peak slopes, is summarized in Table 3.9 on page 109 within the final algorithm comparison section. For each variable, elevation and velocity, the average of the best-fit (peak) slopes from the two error norms is taken to be the best-fit (peak) accuracy of that variable. Plots for each individual algorithm will be shown first, and then all of the algorithms will be compared.

The error norms for the SFD algorithm are presented in Figure 3.1. With the exception of the L_∞ error for the velocity, notice that the slopes for each state variable are roughly the same, as expected from the truncation error analysis. The best-fit (peak) convergence rate for the velocity is 0.914 (1.213), while the elevation rate is 0.983 (1.003).

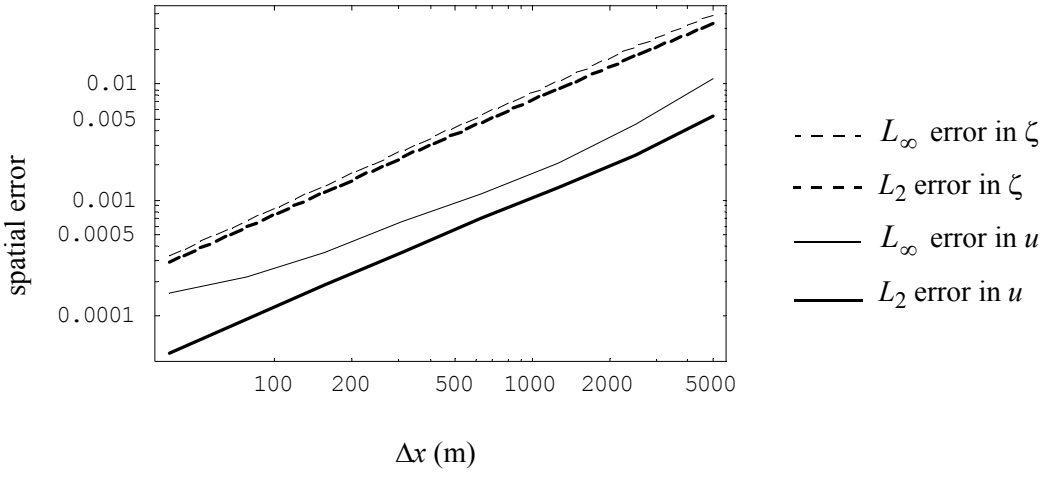


Figure 3.1 Comparison of spatial error norms for the SFD algorithm.

The error norms for the PLF algorithm are presented in Figure 3.2. Notice that as the grid spacing is refined, the convergence rate decreases and the errors eventually level out. Therefore, the best-fit slope is computed using only the linear portion of the curves in order to determine the convergence rates. The best-fit (peak) convergence rates are 1.006 (1.135) for the velocity and 1.464 (1.831) for the elevation.

The error norms for the ADCIRC model (GWC algorithm) are presented in Figure 3.3. The entire range of grid resolutions is used to calculate the slopes; the best-fit (peak) convergence rate for velocity is 1.021 (1.650), while the elevation rate is 1.441 (1.762).

The error norms for the SLFE algorithm are presented in Figure 3.4. Recall that the largest stable time step must be used with each of the grid resolutions in order to maintain a consistent approximation. Table 3.1 summarizes the time step and grid resolution combinations for each of the simulations. Notice that the maximum elevation errors are not significantly different from the average elevation errors, thus the elevation error is uniform over the domain. The best-fit (peak) convergence rate for the velocity is 0.900 (0.960), while the elevation rate is 0.956 (0.990).

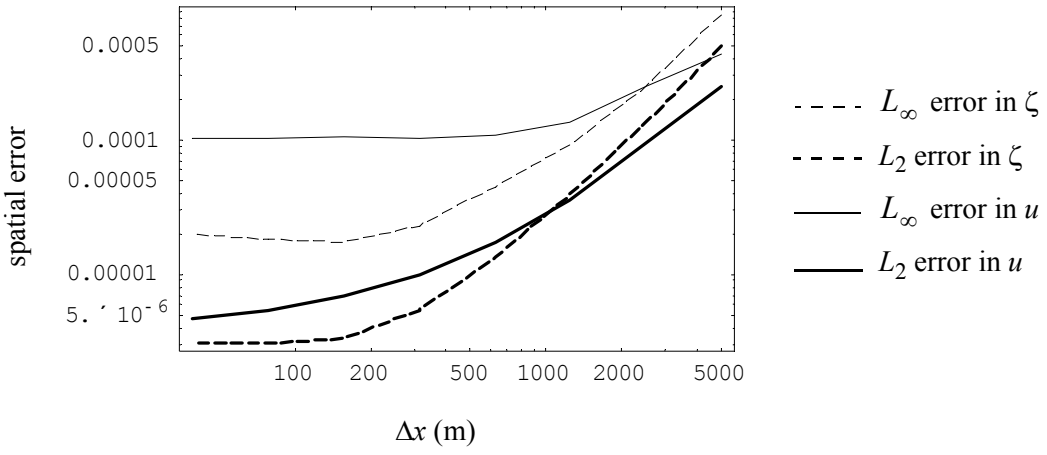


Figure 3.2 Comparison of spatial error norms for the PLF algorithm.

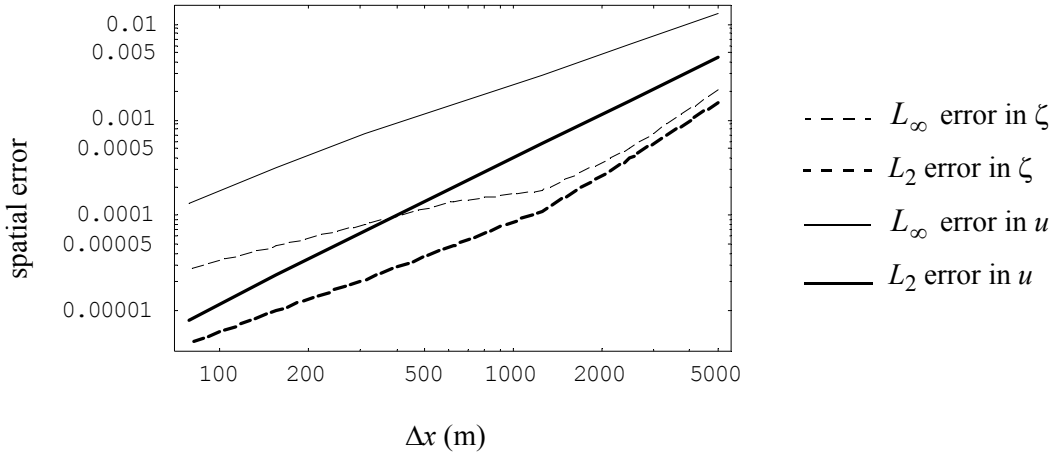


Figure 3.3 Comparison of spatial error norms for the linearized ADCIRC model.

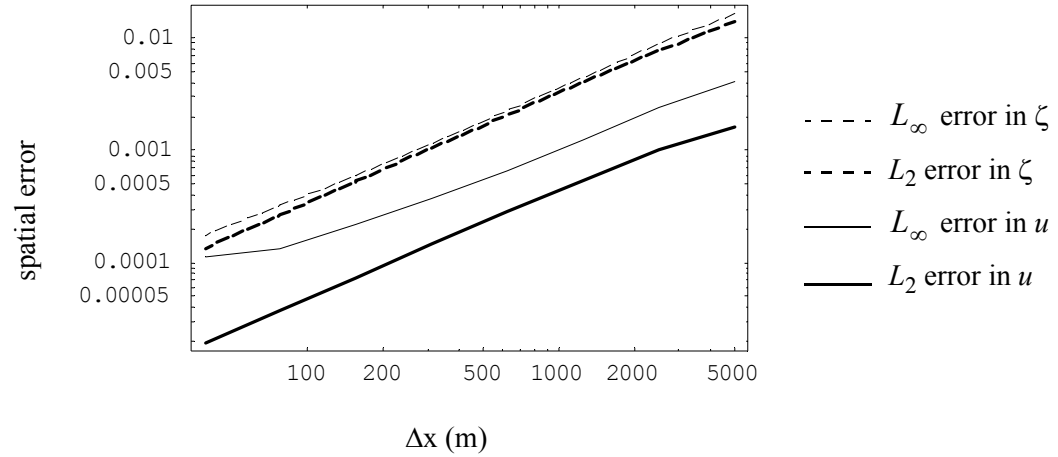


Figure 3.4 Comparison of spatial error norms for the selective lumping FE algorithm.

Table 3.1 Maximum stable time step combinations for SLFE algorithm.

Δx (m)	39.0625	78.125	156.25	312.5	625	1250	2500	5000
Δt (sec)	1.40625	2.8125	5.625	11.25	22.5	45	90	180

If, instead, the small time step of 0.01 seconds is used, the SLFE algorithm starts out with a slower convergence rate and then reaches a steady rate as the grid is resolved. This odd behavior at the coarsest resolutions may be due to the interaction of the spatial and temporal resolution. Due to the small time step, the $\Delta x^2/\Delta t$ terms in the truncation error will likely dominate at the most coarse resolutions.

Figure 3.5 compares the error norms for the low-order finite volume and discontinuous Galerkin algorithms, and the higher-order methods with the three limiters (minmod, superbee, vanLeer). From this graph it is readily apparent that, although the order of the FVM approximations are roughly equivalent, as evidenced by the similar slopes, the

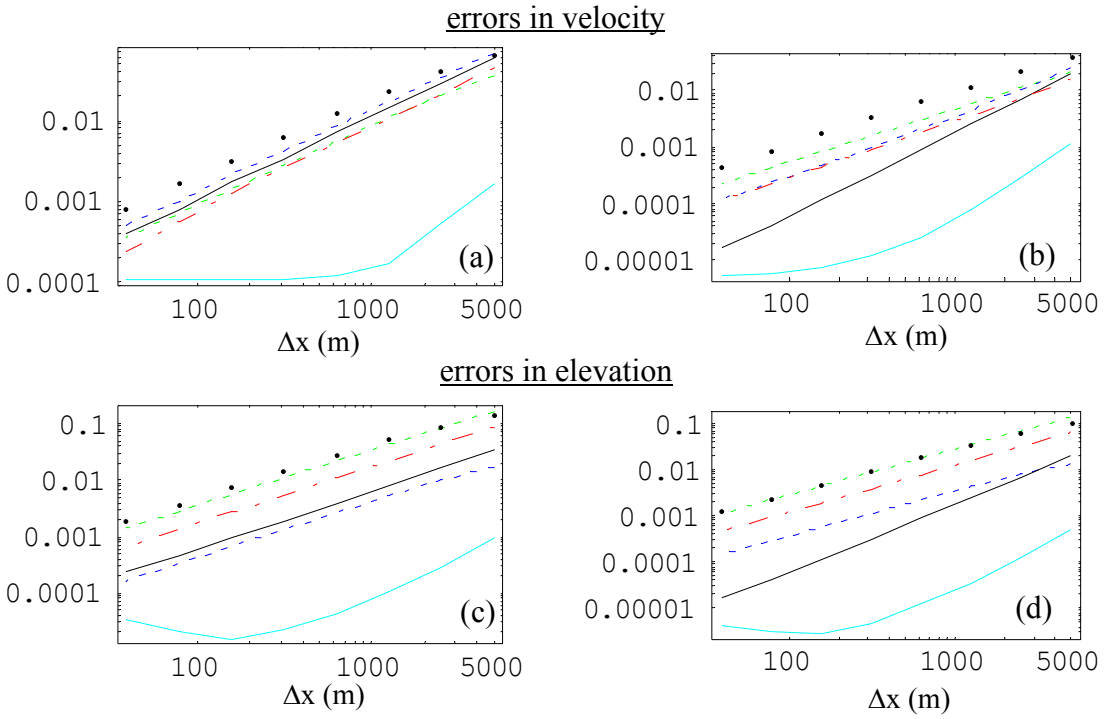


Figure 3.5 Comparison of spatial error norms for the FVM and DG algorithms: (a) L_{inf} for u, (b) L_2 for u, (c) L_{inf} for ζ, (d) L_2 for ζ.

minmod and Superbee limiters are the most accurate of the four, as they have the lowest errors for each of the error norms. Note, however, that the Superbee limiter does exhibit a slightly steeper slope for the L_2 velocity error norm, shown in Figure 3.5b. Notice also that the vanLeer limiter does not provide an improvement in accuracy over the low-order FVM, as its errors lie at or above the low-order curve for all four error measures. The unlimited DG algorithm also has a higher convergence rate, as evidenced by the steeper slope; this algorithm is also the most accurate and has errors significantly lower than the other discontinuous schemes. Meanwhile, the minmod limited DG scheme has similar errors (and convergence rates) to the FVM schemes.

The best-fit convergence rates, computed over the linear range of the curves, and the peak rates for these five algorithms are summarized in Table 3.2. Notice that the convergence rates computed from the L_2 velocity error for the Superbee limited FVM and all four of the error norms for the DG unlimited algorithm are higher than the remaining entries in the table. Note also that the unlimited DG scheme has significantly higher convergence rates than any of the other algorithms, including the minmod DG scheme.

Table 3.2 Best-fit linear (peak) convergence rates for discontinuous algorithms.

Figure 3.5 panels	(a) L_∞ for u	(b) L_2 for u	(c) L_∞ for ζ	(d) L_2 for ζ
low-order FVM/DG	0.965 (0.998)	0.938 (0.970)	0.987 (1.005)	0.994 (1.001)
FVM – minmod	1.016 (1.090)	1.098 (1.277)	0.968 (1.016)	0.947 (0.992)
FVM – Superbee	1.034 (1.118)	1.466 (1.544)	1.032 (1.123)	1.002 (1.013)
FVM – vanLeer	0.943 (0.991)	0.938 (0.978)	0.940 (0.992)	0.951 (0.995)
DG – unlimited	1.651 (1.679)	1.666 (1.921)	1.338 (1.696)	1.682 (2.030)
DG – minmod	1.067 (1.290)	1.012 (1.131)	1.008 (1.028)	1.012 (1.054)

Although the unlimited DG algorithm is still stable at this time step, it is beginning to lose accuracy in the elevation results as the grid is refined past $\Delta x = 156.25$m. (At a larger time step of 0.1 seconds, the unlimited higher-order DG becomes unstable at the finest resolution.) Also, the unlimited higher-order DG algorithm, which has the linearity of the

functions built-in to the approximation, has much better accuracy than the various limited FVM choices, which use ad hoc post-processing to add the higher-order terms. Therefore, it would appear that the higher-order DG method is more accurate than the equivalent FVM, although it is also more sensitive to stability limits.

A comparison of the spatial error norms for the SFD, PLF, ADCIRC, SLFE, low-order FVM/DG, minmod limited FVM, and minmod limited DG algorithms is given in Figure 3.6. Notice that the PLF and unlimited DG algorithms have the lowest errors, and also that there convergence plots are very similar. The unlimited DG is more accurate than the ADCIRC model, although this is more significant for the velocities than the elevations. Additionally, this reduction in error is only significant enough (to justify the extra computational effort of the higher-order DG model) at the larger grid resolutions.

Notice that the minmod limited FVM and the SLFE algorithms have similar error behavior for the elevation results, but that the velocity results differ significantly. Also, although the truncation error analysis predicts higher theoretical convergence rates for some of the algorithms, the figure indicates that most of the algorithms have similar rates (slopes) in practice. Only the ADCIRC, PLF and unlimited DG have steeper slopes in select regions of the convergence plots. This will be discussed further when the accuracy is compared in tabular form in Table 3.9 on page 109.

3.4.2 Temporal accuracy

For the temporal convergence studies, interval halving with a base time step of Δt_b = 1.6 seconds was used to evaluate the accuracy of the algorithms, such that the errors are computed for two successive solutions with time steps of $\Delta t_b/m$ and $\Delta t_b/2m$. Each simulation uses a time step one half as large as the previous simulation (interval halving), until the ratio of successive error norms converge to a constant. This converged ratio represents the accuracy of the algorithm. In order to isolate the temporal errors, a fixed spatial grid with 640 elements was used for all of the simulations.

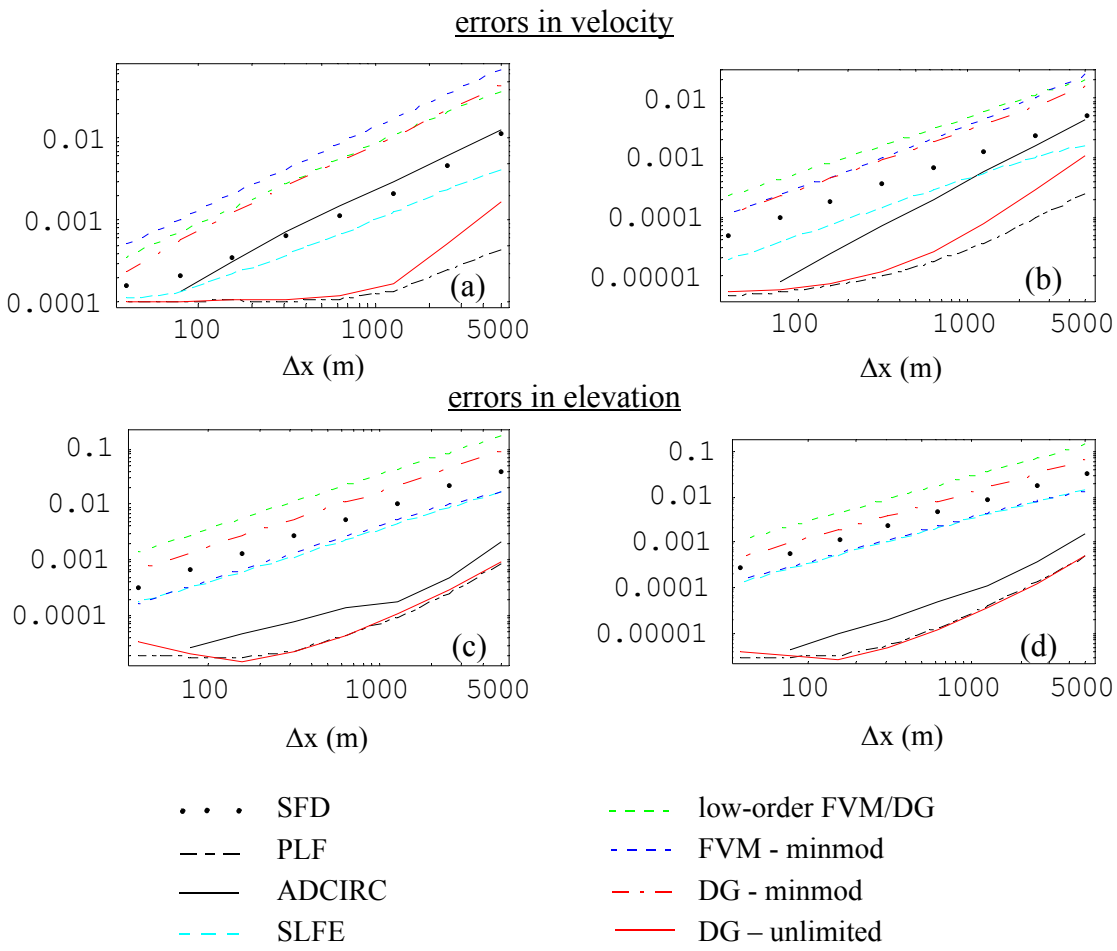


Figure 3.6 Comparison of spatial error norms for all study algorithms: (a) L_{inf} for u, (b) L_2 for u, (c) L_{inf} for ζ, (d) L_2 for ζ.

A graphical presentation of the results provides for easier comparison between algorithms, therefore only the log-log plots of error versus m will be presented in this section. The tabular data used to generate these plots is available to the interested reader in Appendix D. Also, the average of the converged temporal accuracy for the L_∞ and L_2 error norms for each state variable, elevation and velocity, are reported in Table 3.9 on page 109 within the final algorithm comparison section. Plots for each individual algorithm will be shown first, and then all of the algorithms will be compared.

The error norms for the SFD algorithm are presented in Figure 3.7. Notice that the

elevation and velocity errors are equivalent. Both error norms for the two state variables converge toward zero as the time step is cut in half successive times and the order of accuracy is similar for all four curves. The converged ratio between m=8 and m=16 gives the accuracy as 1.941 for velocity and 1.950 for elevation.

The error norms for the PLF algorithm are presented in Figure 3.8. Notice that the elevation errors are slightly higher than the velocity errors. The converged ratio between

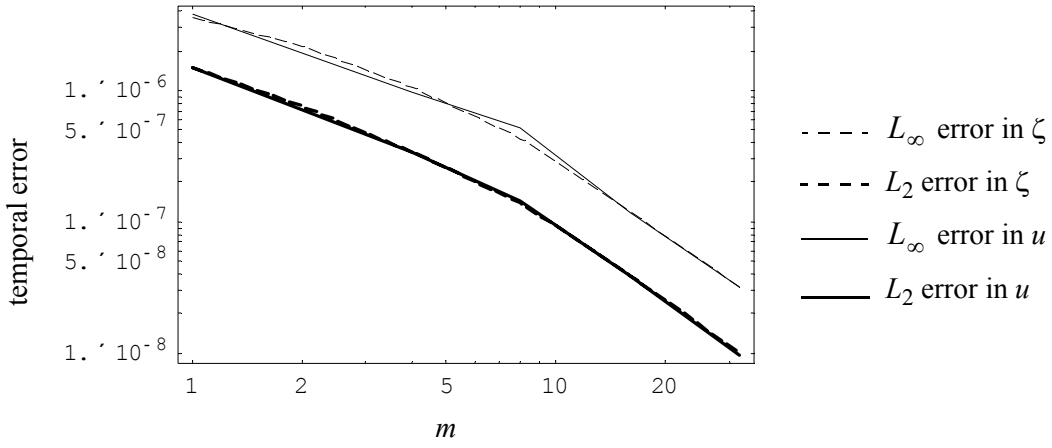


Figure 3.7 Comparison of temporal error norms for the SFD algorithm.

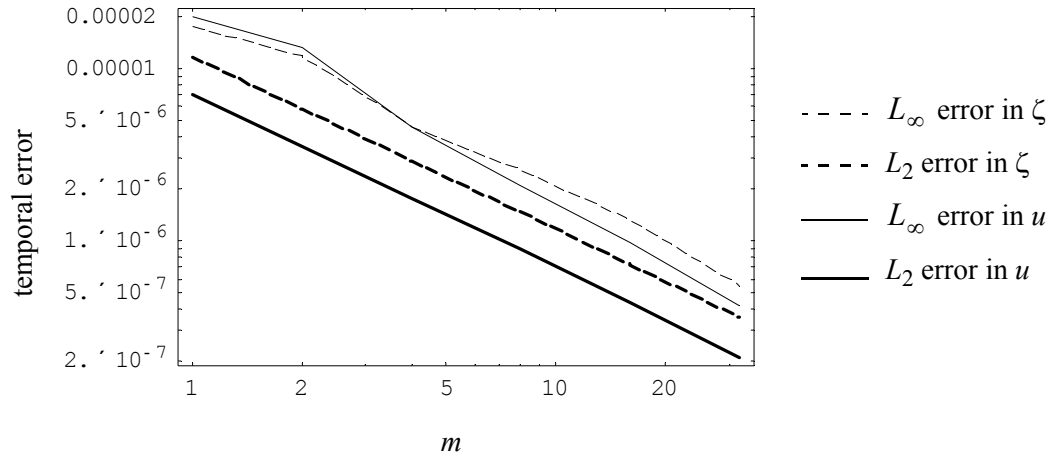


Figure 3.8 Comparison of temporal error norms for the PLF algorithm.

m=16 and *m*=32 gives the accuracy as 1.133 for velocity and 1.142 for elevation.

The error norms for the ADCIRC model are presented in Figure 3.9. Notice that the elevation and velocity errors are similar through *m*=4, at which point the velocity errors begin to increase. At m=16, all four error norms begin to increase, such that the solution appears to be diverging. (If one takes another division of the time step – $m = 64$, $\Delta t = 0.0125$ seconds – the errors continue to increase.) When a smaller grid spacing is used (Δx = 39.0625m), the errors come back into line; although the slope of the lines for the smaller time steps are shallower than for the larger time steps. The converged ratio between *m*=2 and *m*=4 gives the accuracy as 2.007 for velocity and 1.952 for elevation.

When looking at the spatial distribution of the temporal errors for the ADCIRC model between successive simulations, the errors are highest at the boundaries. Since the elevation BC is enforced at the ocean boundary and the velocity BC is enforced at the land boundary, the elevation errors are highest at the land boundary and vice versa for the velocity errors. This same problem was noted in other studies by *Dresback and Kolar* [2004a] and *Dresback et al.* [2005], where the spatial errors converge without incidence but the temporal errors do not converge as Δt is refined past a certain point. However, the model

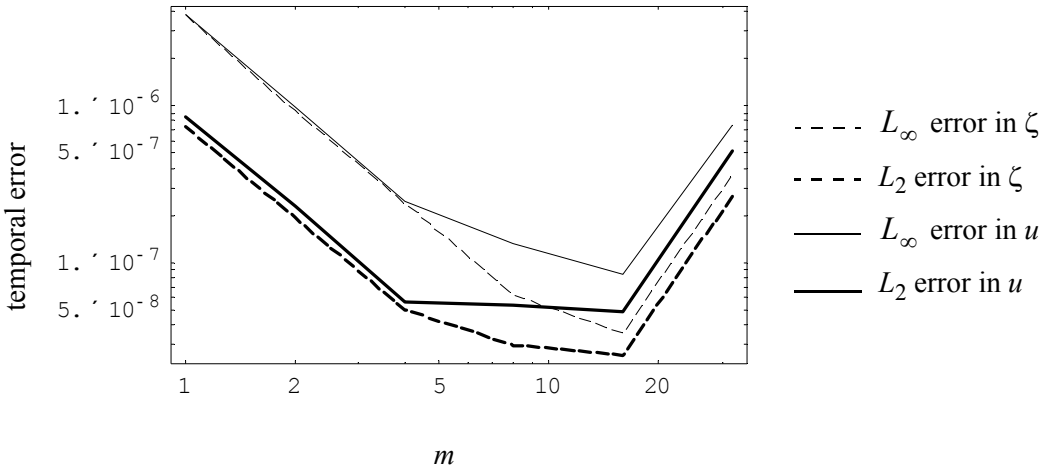


Figure 3.9 Comparison of temporal error norms for the ADCIRC model.

remains accurate on the interior as Δt is refined. Intuitively, as more time steps are required to reach the same simulation time when using smaller values of Δt, the errors will begin to accumulate more at the opposite boundaries as the time step is refined further.

The temporal error norms for the SLFE algorithm ($e = f = 0.9$) are presented in Figure 3.10. Recall that the base time step is roughly twice as large as for the other algorithms (2.8125 versus 1.6 seconds); thus to have an equal comparison with the other algorithms later, m starts at 1/2 for this model.

With the exception of the beginning of the elevation error curves, this algorithm behaves opposite of what would be expected of a general algorithm in that the errors increase as the time step is cut in half successive times. *Kawahara et al.* [1982] note that the largest stable time step must be used in order to minimize the numerical damping. However, as discussed in the prior section on truncation error (§3.3.4), the larger Δt is needed to keep the algorithm consistent with the continuum equations. The truncation error analysis showed that for values of e other than 1, the second term of the error is $O(\Delta x^2 / \Delta t)$, which would increase as the time step is decreased. The converged ratio between m=8 and m=16 gives the accuracy as –0.987 for velocity and –0.683 for elevation,

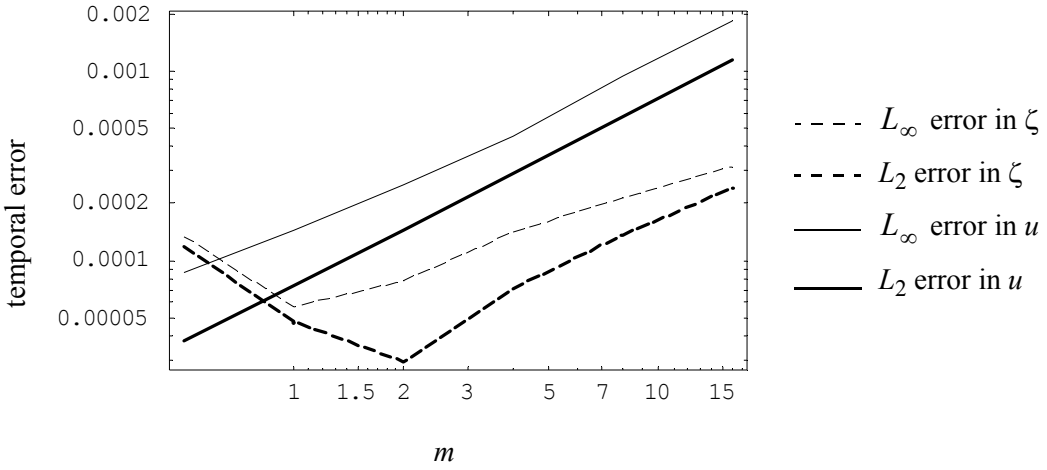


Figure 3.10 Comparison of temporal error norms for the SLFE algorithm.

where the negative signs indicate that the slope is opposite of what should be expected.

Figure 3.11 compares the error norms for the low-order finite volume and discontinuous Galerkin algorithms, and the higher-order versions with the studied limiters (minmod, superbee, vanLeer). From this graph it is readily apparent that the temporal convergence rate of the FVM and DG approximations are roughly equivalent, as evidenced by the similar slopes as the curves converge at higher m values. The superbee limited FVM and unlimited DG curves begin at $m = 16$ ($\Delta t = 0.1$ sec) because they are unstable for larger time steps at this grid resolution. Notice, also that the minmod limited FVM and DG models have nearly equivalent convergence plots and remain stable over the range of time steps

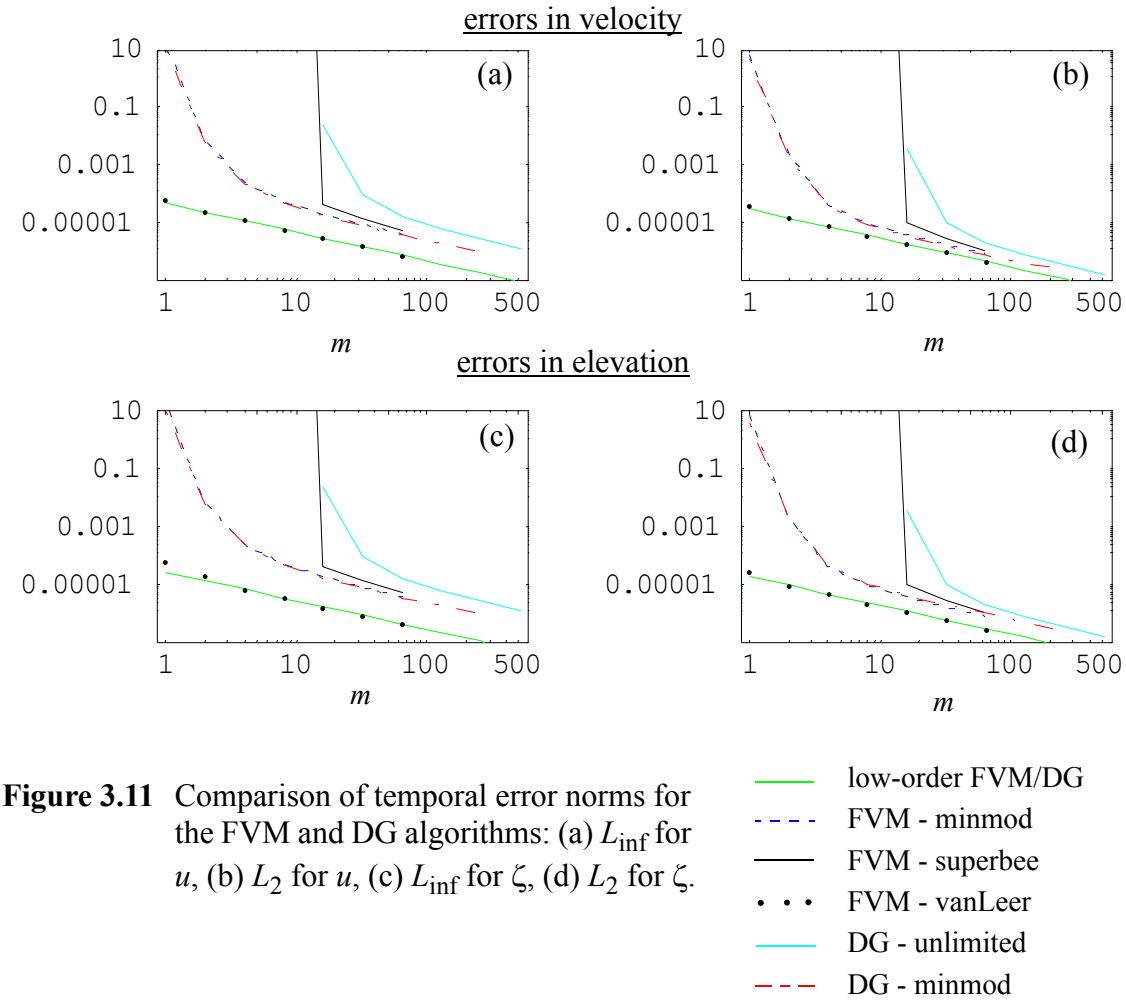


Figure 3.11 Comparison of temporal error norms for the FVM and DG algorithms: (a) L_{inf} for u, (b) L_2 for u, (c) L_{inf} for ζ, (d) L_2 for ζ.

used in this study, but that they appear to be approaching instability at a time step of Δt = 1.6 seconds.

Somewhat surprisingly, the low-order FVM/DG algorithm has lower temporal errors than the higher-order slope limited algorithms. However, the high-order nature of these algorithms lies in the spatial realm, and the first-order explicit time stepping scheme does not allow them to outperform the low-order spatial schemes in the temporal realm. In practice, a temporal scheme on the same order as the spatial approximation is typically employed with discontinuous algorithms (e.g., second-order Runge-Kutta for linear spatial approximations).

As noticed in the spatial grid convergence study, the superbee limited FVM and the unlimited higher-order DG algorithms are more susceptible to stability issues. In addition these algorithms have the highest errors of all the discontinuous schemes. Also, as was the case in the spatial convergence study, the vanLeer limited FVM algorithm has the same error characteristics as the low-order FVM/DG algorithm. For future work, it would be interesting to re-run the higher-order discontinuous models with second-order Runge-Kutta time stepping to compare the temporal convergence.

The converged ratios and resulting accuracy values for each of the algorithms are summarized below in Table 3.3. Note that the rates for the Superbee limited FVM algorithm may be artificially high, since the ratios may not have converged yet by m=62.

Table 3.3 Temporal convergence rates for discontinuous algorithms.

Figure 3.11 panels	(a) L_∞ for u	(b) L_2 for u	(c) L_∞ for ζ	(d) L_2 for ζ
low-order FVM/DG	1.001	1.000	1.000	1.001
FVM - minmod	1.086	1.052	1.095	1.060
FVM - Superbee	1.255	1.271	1.272	1.288
FVM - vanLeer	1.000	1.000	1.000	1.001
DG - unlimited	1.078	1.056	1.080	1.059
DG - minmod	0.993	0.979	0.997	0.987

Otherwise, all of the discontinuous schemes have first-order temporal accuracy, as is expected for a simple forward-Euler temporal discretization.

A comparison of the temporal error norms for the SFD, PLF, ADCIRC, SLFE, low-order FVM/DG, minmod limited FVM and DG, and unlimited DG algorithms is given in Figure 3.12. The ADCIRC model has the lowest temporal errors followed closely by the SFD then by the PLF, low-order FVM/DG, minmod limited FVM and DG, unlimited DG, and SLFE algorithms. Again, notice that the SLFE convergence curve goes in the opposite direction of all of the others since the maximum stable time step provides the most accurate

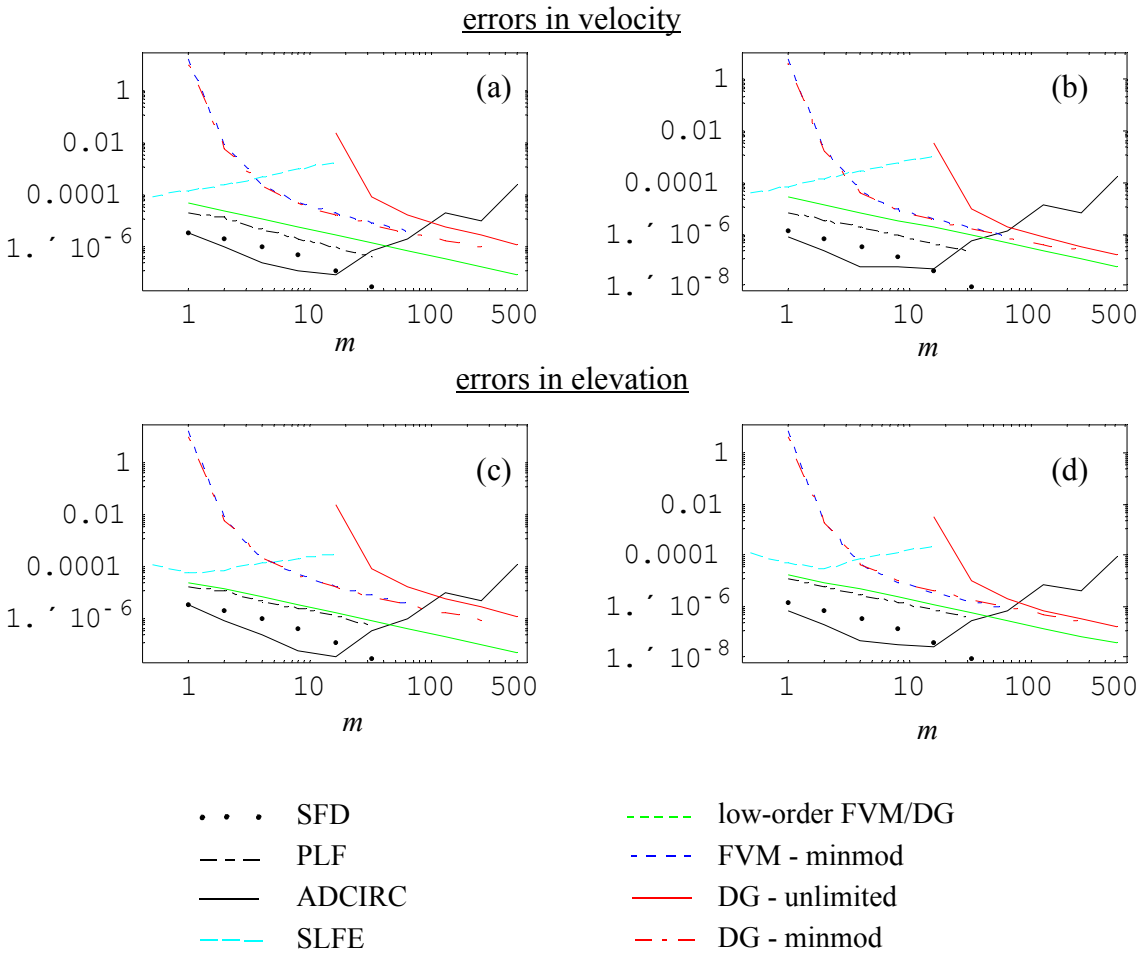


Figure 3.12 Comparison of temporal error norms for all study algorithms: (a) L_{inf} for u, (b) L_2 for u, (c) L_{inf} for ζ, (d) L_2 for ζ.

results for this algorithm. Also, recall from the individual algorithm results, that the ADCIRC model begins to accumulate error near the boundaries as the time steps are reduced past m = 16 (Δt = 0.1 sec). The unlimited DG curve begins at m = 16, as the algorithm is unstable at higher time steps, and goes out to higher m values to get closer to a converged ratio. From these figures, it appears that the ADCIRC model has a slightly higher convergence rate than the other algorithms (for the linear converging portion of the plot from m = 1 through 16). A more precise comparison is made when the numerical accuracy results are compared with the analytical predictions in Table 3.9 on page 109.

3.5 Analytical propagation behavior for the SWE algorithms

The results of the analytical propagation behavior using Fourier and dispersion analyses are presented herein. Stability limits are also derived for each algorithm using the Fourier propagation factors.

3.5.1 Discussion of wave propagation behavior

The Fourier propagation characteristics and the dispersion curves for the SFD, PLF, ADCIRC, SLFE and low-order FVM/DG algorithms, as derived from the equations presented in Chapter 2, are shown below in Figure 3.13. The parameter values used to generate these curves are the same as those used in Chapter 2, which are repeated here for convenience in Table 3.4. However, the equations for these curves, which were also presented in Chapter 2, will not be repeated here. Note that an analytical dispersion curve for the SLFE is not provided since the time stepping is an integral part of the algorithm. Also, two time steps are examined for the Fourier propagation characteristics of the SLFE algorithm: Δt = 0.1 and 50 seconds, since the maximum stable time step must be used in order to obtain optimal (consistent) results with this algorithm. Additionally, no analytical results are shown for the higher-order FVM and DG algorithms since it is not possible to write them in the necessary closed form, when slope limiters are used.

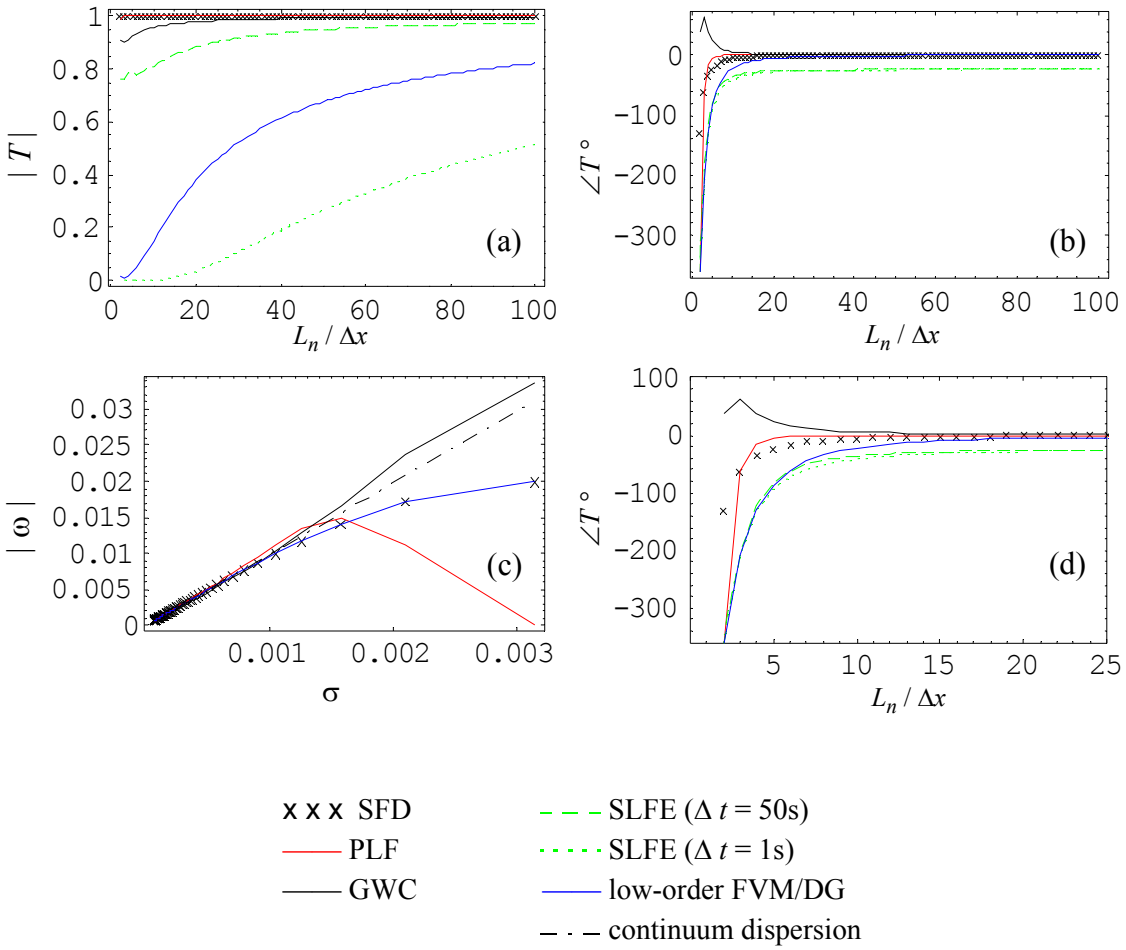


Figure 3.13 Comparison of propagation characteristics for the study algorithms: (a) Fourier damping ratio, (b) Fourier phase error, (c) dispersion relationship, (d) zoomed-in view of Fourier phase error.

Table 3.4 Parameters used in propagation study.

$g = 9.81$ m/s^2	$h = 10$ m	$\Delta t = 1.0$ s
$\tau = 0.0001$ s^{-1}	$G = 0.001$s^{-1}	$\Delta x = 1000$ m
$e = 0.9$	$f = 0.9$	$\Delta t = 50.0$ s

For the Fourier propagation characteristics notice that the PLF, SLFE and low-order FVM/DG algorithms all have a phase lag of 360 degrees for the $2\Delta x$ wavelength. Of these three algorithms, however, the PLF is the only one that does not have any damping ($|T| = 1$), which implies that if this wavelength is introduced into a simulation, it will not propagate or be damped out and will thus overlie the correct solution. This is also evident in the dispersion plot for this algorithm where the folded curve indicates that for each temporal frequency there will be two spatial frequencies (high-frequency noise superimposed on the physical long-wave solution). The low-order FVM/DG algorithm completely damps this high frequency noise out, as evidenced by the damping ratio of 0.0, but it is also over-dissipative in the longer, physical wavelengths (the damping ratio never reaches unity although the phase error approaches zero). The SLFE algorithm is also more dissipative than the other algorithms, and it reaches an asymptotic value of -24 degrees for the phase error, so that it is always out of phase even in the physical wavelengths. Notice also that the phase error for the SLFE algorithm does not differ significantly for the two different time steps; however, the smaller time step (Δt = 0.1 seconds) results in considerably more damping.

The SFD algorithm has a phase lag in the low wavelengths ($2 - 20\Delta x$) with very little to no damping, however the lag is not large enough to cause the dispersion curve to fold completely, so it is still monotonic. Interestingly, the dispersion curve for this algorithm is identical to that of the low-order FVM/DG algorithms; yet the Fourier propagation characteristics are quite different. This is due to the similarity of the spatial discretizations (element-centered elevations) for the two algorithms. As the dispersion behavior is derived using continuum time, the spatial discretization alone determines the propagation relationship. However, the Fourier propagation characteristics take into account the temporal discretization of the algorithms, which results in the difference in Fourier behavior, despite the identical dispersion curve.

The PLF algorithm has a phase lag for the $2\Delta x$ wavelength but then changes to a

slight phase lead for the next several wavelengths. The GWC algorithm begins with a noticeable phase lead (+60 degrees) at the small wavelengths, which decays to a slight phase lead (on the order of 5 degrees) for the remainder; however there is sufficient damping in the GWC algorithm to avoid wiggles. Notice that the dispersion curves for these two algorithms lie above the continuum curve for wavenumbers within this range, where there is a phase lead.

From these plots, it is evident that the PLF algorithm does not have good propagation properties for the low wavelengths, the SLFE algorithm always exhibits a significant phase error for even for the physical wavelengths, and the low-order FVM/DG algorithms are overly dissipative. What is not shown in these plots is that the SLFE algorithm is extremely sensitive to the time step and lumping parameter, the subject of the next section.

3.5.2 Parameter study for SLFE algorithm

In this section, a brief parameter study of the SLFE algorithm is presented in the context of Fourier propagation characteristics. Recall that the SLFE algorithm, as presented by *Kawahara et al.* [1982], uses a selective lumping parameter, e, to determine whether the mass matrices are lumped, unlumped or partially lumped. In this study, we have introduced a separate lumping parameter, f, for the bottom friction terms. For a fixed time step, Δt, and general lumping parameter, e, the following observations are made as the bottom friction lumping parameter is varied from unlumped (f=0) to fully lumped (f=1):

(*i*) the phase error is the same over the entire range of values for the bottom friction lumping parameter (this is true for any combination of e and Δt, although the results do differ as e or Δt are changed);

(*ii*) when the value of e is near 1.0, the damping ratio is slightly better when the bottom friction term is unlumped, but as the value of e decreases, the damping ratio is not significantly altered as f varies over the entire range of values;

(*iii*) following from points (*i*) and (*ii*), for any fixed value of *e* (over the entire range 0 to 1) as Δt increases, all of the phase error and damping ratio curves for the range of *f* values converge to the same value.

From these observations, the differences in the propagation results as *f* varies from unlumped to fully lumped are not significant. Therefore, it is recommended that a single lumping parameter be used by setting *f* equal to *e*.

Now, given a fixed value for the lumping parameters with $e = f$, the following observations are made as the time step is varied:

(*i*) in general the damping ratio and phase error improve as Δt increases until it nears the maximum stable time step, after which the results begin to deteriorate again;

(*ii*) for values of the lumping parameter near 0.0, the propagation characteristics are quite bad for any time step (when $e = f = 0.0$ and the maximum stable time step is used, phase errors approach -240 degrees for the long, physical wavelengths and damping ratios approach 0.8 for these same wavelengths, while the phase error starts out at -360 degrees but the damping ratio is 0.05);

(*iii*) for values of the lumping parameter at 1.0 and any stable time step, the propagation characteristics are similar to the PLF algorithm, namely the phase error starts out at -360 degrees and approaches 0.0 but the damping ratio is always 1.0 so that short wavelength noise does not propagate nor dissipate;

(*iv*) values of the lumping parameter near 1.0 (>0.8) minimize the asymptotic phase error for the physical wavelengths but also reduce the damping of the lagging short wavelength noise, while lower values of the lumping parameter (<0.5) provide more damping for the short wavelength noise but also damp out the physical waves, which have greater phase lags;

(*v*) for any lumping coefficient, as the time step decreases away from the maximum stable

value, the damping ratio decreases significantly while the phase error changes only slightly in the low wavelength range ($< 20\Delta x$) and remains unchanged in the physical wavelength range ($> 20\Delta x$).

From these observations, the range of lumping parameter values suggested by *Kawahara et al.* [1982], $0.8 < e < 0.95$ seems to be a sensible balance of these issues. At the lower end when $e = f = 0.8$, the phase error starts at -360 degrees and approaches a value of -50 degrees while the damping ratio starts at 0.57 and approaches 0.97. At the upper end when $e = f = 0.95$, the phase error starts at -360 degrees and approaches a value of -12 degrees while the damping ratio starts at 0.88 and approaches 0.99. Obviously, when the algorithm nears being fully lumped the phase error decreases; however, so does the damping, which could lead to spurious oscillations. Additionally, their recommendation that the largest stable time step be used, in order to reduce the artificial damping, can be confirmed by these propagation results (and is necessary given the results from the truncation error study for this algorithm given in §3.3.4), as will be shown in the following example.

For the value of the lumping parameters used in the previous section (§3.5.1), $e = f = 0.9$, plots of the damping ratio and phase error, as the time step is varied, are shown in Figure 3.14. The smaller time steps are shown in shades of red, while the larger time steps are shown in blue, where the steps of interest are $\Delta t = [0.1, 1, 5, 10, 25, 50, 75]$ seconds. Notice that the domain of the damping ratio is extended relative to the phase error, since the phase reaches its asymptotic value at a smaller wavelength.

As discussed above, notice that despite the large range of values used for the time step, the phase error is only slightly different in the small wavelength range, L_n from 2 to $20\Delta x$, and exhibits no variation at all for the physical wavelengths. Also, the phase error never reaches zero, but instead reaches an asymptotic value around -25 degrees. Notice also that the smaller time steps (red curves) severely overdamp the solution, while the larger

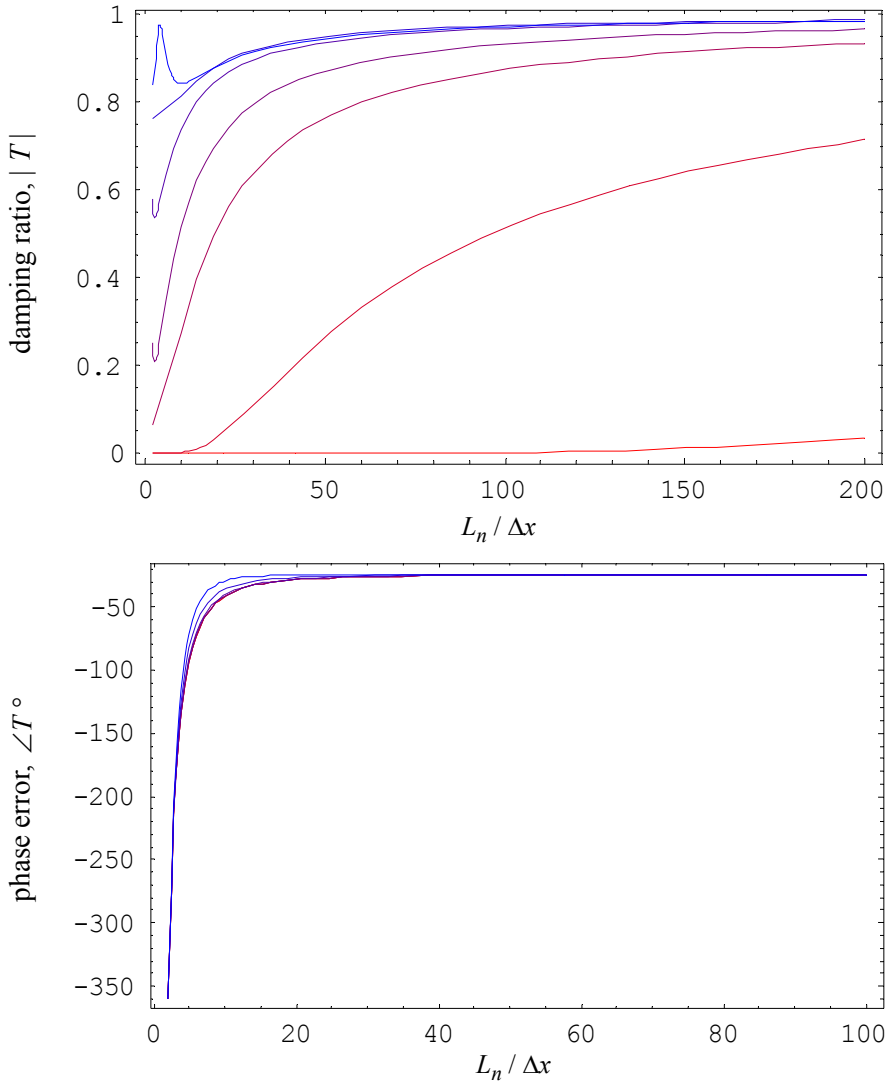


Figure 3.14 Fourier propagation results for the SLFE algorithm with lumping parameter $e = f = 0.9$ and Δt = [0.1 to 75.0] seconds.

time steps (blue curves) do not reach unity but are not as overdamping. The time step of 75 seconds is heading towards instability at the $3\Delta x$ wavelength, as evidenced by the ridge in the damping ratio. (As discussed in §2.2.1, the damping ratio, $|T|$, can be greater than unity without the algorithm being unstable; however, examination of $|\lambda|$ indicates that the algorithm is indeed approaching instability.) In order to avoid this strange behavior at the $3\Delta x$ wavelength, a value of $\Delta t = 50$ seconds was chosen as the "maximum stable time step"

for the propagation study presented above in §3.5.1.

3.5.3 Stability from Fourier propagation factors

Due to the complexity of the algebraic equations, it was not possible to obtain a closed-form limit for all of the algorithms from their respective propagation factors. For those that lent themselves to such a solution, the relationship for the maximum stable time step is given. For the remaining algorithms, a limit was found numerically by calculating $|\lambda|$ for all wavelengths for successively larger time steps until this magnitude reaches or exceeds unity. Even these computed limits will be referred to as analytical limits since they are derived from the analytical expressions for the propagation factors and not from numerical simulations. All of the propagation factors are functions of the wavenumber, $\sigma = 2\pi/L_n$; for most algorithms, the most critical wavelength is between $L_n = 2\Delta x$ and $5\Delta x$. For the algorithms where a closed expression is obtained, the critical wavelength was first determined numerically and then the algebraic equations for this wavelength were solved. These analytical stability limits are compared to those obtained from numerical simulations in the summary section, §3.7.

Staggered finite difference

For the SFD algorithm, the critical wavelength is $L_n = 2\Delta x$, which results in the following expression for the propagation factor:

$$\lambda = \frac{16gh\Delta t^2(1-\theta)\theta + \Delta x^2\left(4 + 2\tau\Delta t(2\theta-1) \pm \frac{2\Delta t}{\Delta x}\sqrt{-16gh + \tau^2\Delta x^2}\right)}{4(4gh\Delta t^2\theta^2 + \Delta x^2(1+\tau\Delta t\theta))}. \qquad \textbf{(3.45)}$$

When $|\lambda| \le 1$, this gives the following constraints on Δt:

$$\Delta t > \frac{\tau\Delta x^2(1-2\theta) \pm \Delta x(1-2\theta)\sqrt{-16gh + \tau^2\Delta x^2}}{2(2gh - 8gh\theta + 8gh\theta^2)}. \qquad \textbf{(3.46)}$$

For Crank-Nicolson time stepping, θ = 0.5, which results in a single constraint: $\Delta t > 0.0$ seconds. This is always true, and the analysis indicates that the algorithm should be unconditionally stable. Numerically, the propagation factor was found to be less than or equal to 1.0 for any time step (and wavelength), verifying that the algorithm is unconditionally stable for Crank-Nicolson time stepping.

Primitive leap-frog

For the PLF algorithm, the critical wavelength is $L_n = 3\Delta x$, which results in the following expression for the propagation factor:

$$\lambda^2 = 1 - 6gh\left(\frac{\Delta t}{\Delta x}\right)^2 - \tau\Delta t \pm 2i\sqrt{3gh\left(\frac{\Delta t}{\Delta x}\right)^2 - \left(3gh\left(\frac{\Delta t}{\Delta x}\right)^2 + \frac{\tau\Delta t}{2}\right)^2}. \quad \textbf{(3.47)}$$

From this, the stability constraints are then given as

$$\Delta t \geq \frac{-\tau\Delta x^2 - \sqrt{12gh\Delta x^2 + \tau^2\Delta x^4}}{6gh} \text{ and} \quad \textbf{(3.48)}$$

$$\Delta t \leq \frac{-\tau\Delta x^2 + \sqrt{12gh\Delta x^2 + \tau^2\Delta x^4}}{6gh}. \quad \textbf{(3.49)}$$

For the parameter values used in this study (given in Table 3.4), these constraints are calculated to be $\Delta t \geq -58.46$ and $\Delta t \leq 58.12$. The first constraint is always satisfied since Δt is always greater than zero.

ADCIRC model (GWC)

For the ADCIRC model, the critical wavelength is $L_n = 5\Delta x$. The third order polynomial for the propagation factor produces three roots, all of which are too complicated to produce a closed-form expression for the propagation factor. Therefore, a numerical solution of the algebraic equations for the stability limit was computed instead. For $|\lambda| \leq 1$

the maximum time step is $\Delta t = 100.08$ seconds.

Selective lumping finite element

For the SLFE algorithm, the critical wavelength is $L_n = 4\Delta x$. Again, the expression for the propagation factor is too complicated to present here and a numerical solution of the algebraic equations for the stability limit was computed instead. For $|\lambda| \leq 1$ the maximum time step is $\Delta t = 79.23$ seconds.

Low-order finite volume / discontinuous Galerkin

For the low-order FVM/DG algorithms, the critical wavelength is $L_n = 2\Delta x$, which results in the following expression for the propagation factor:

$$\lambda = \frac{1}{2}\left(2 - 4a\frac{\Delta t}{\Delta x} - \tau\Delta t \pm \tau\Delta t\right). \quad \textbf{(3.50)}$$

From this result, the stability constraints are then found to be

$$\Delta t \leq \frac{\Delta x}{a} \text{ for the negative root and} \quad \textbf{(3.51)}$$

$$\Delta t \leq \frac{2\Delta x}{2a + \tau\Delta x} \text{ for the positive root.} \quad \textbf{(3.52)}$$

Given the parameter values used in this study (Table 3.4), the constraints are calculated to be $\Delta t \leq 100.96$ for the negative root and $\Delta t \leq 100.46$ for the positive root, where both constraints are satisfied as long as the most restrictive case for the positive root is satisfied.

3.6 Numerical propagation behavior for the SWE algorithms

The numerical propagation analysis techniques from Chapter 2 are used to analyze higher-order (piecewise linear) FVM and DG discretizations, wherein the slope limiting procedures prohibit a traditional analytical analysis, and the SLFE algorithm, which does

not lend itself to analytical dispersion analysis due to the two-level time stepping scheme. The numerical propagation techniques that were recommended at the end of Chapter 2 are carried forward for these analyses. Namely, initialization method 2 (IVP with wave near center of domain) and least square harmonic (LSH) wave tracking will be used for Fourier characteristics, while initialization method 3 (BVP with wave introduced through boundary forcing) and both LSH and peak wave tracking will be used for dispersion behavior.

3.6.1 Higher-order finite volume method

For the FVM, the three slope limiters presented earlier in Section 3.2.5 – minmod, vanLeer and Superbee – are analyzed. The propagation curves, as generated by the numerical techniques, are presented in Figure 3.15; notice that all of the phase and

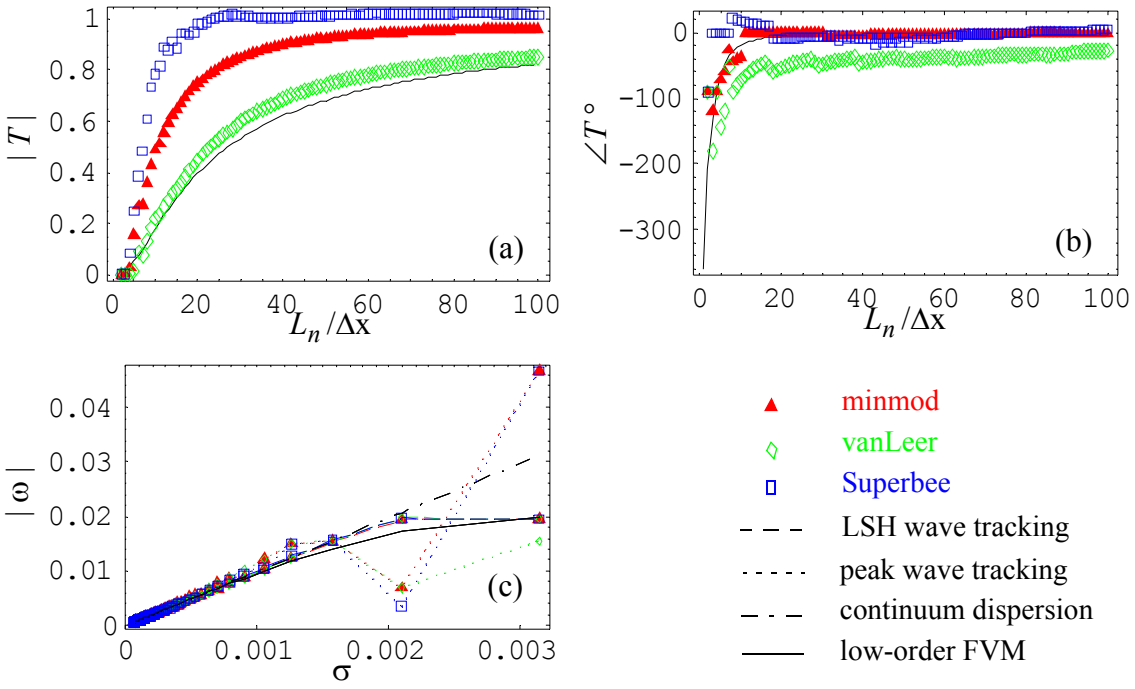


Figure 3.15 Propagation characteristics of higher-order FVM methods using slope limiters: (a) Fourier damping ratio, (b) Fourier phase error, and (c) dispersion curve.

magnitude data are discrete since there are no analytical results to present, but that the continuum dispersion behavior and low-order results have been plotted for reference.

The minmod slope limiter scheme exhibits a slight phase lag in the low wavelengths, but these would be damped out, as shown in the magnitude plot. However, this scheme appears to be overly dissipative, as the magnitude only reaches a maximum value of approximately 0.9, implying that the long physical waves will also be damped, despite their correct phase. The vanLeer limiting scheme exhibits an asymptotic phase lag where the long wavelength, physical waves are approximately 25 degrees out of phase, and they reach a relative magnitude of only 0.8 of the continuum waves. Thus, this scheme would not appear to be a good choice in practice. In contrast to these two limiting methods, the Superbee exhibits good phase behavior and much less damping, as it approaches the desired magnitude of 1.0 within the first $20\Delta x$ wavelengths. This is within the common range used in application (i.e., in practice waves smaller than $20\Delta x$ are not usually simulated, but rather the grid spacing is adjusted accordingly).

The dispersion results (Figure 3.15c) indicate that all three limiters remain monotonic but exhibit a slight folding away from the continuum, when LSH wave tracking is utilized for the propagation analysis. Meanwhile, if peak tracking is used to monitor the position of the waves, then there is a dip corresponding to the $3\Delta x$ wavelength, while the wavenumber corresponding to the $2\Delta x$ wavelength lies above the continuum for the minmod and Superbee limiters and the vanLeer dispersion curve remains folded. Given the phase lags seen in Figure 3.15b and the low-order behavior, the LSH dispersion plots seem more reasonable. Considering both Fourier and dispersion results, the Superbee class of limiters appears to be the most promising.

Note that even with the higher-order FVM, the additional second-order spatial derivative introduced via the Riemann problem still exists (see §3.3.5 for discussion). However, if one is judicious about which slope limiting algorithm is used to reconstruct the

slopes, only enough diffusion is added to keep the algorithm stable ("controlled" diffusion) is added. From the results in Figure 3.15, it is apparent that the Superbee class of limiters is more successful than the other two at controlling the amount of dissipation that is introduced by the second-order spatial derivative. Thus, with the numerical propagation results, there now exists a tool to judge slope limiters for their ability to damp noise without damping the physical waves as well.

Figure 3.16 compares 1D elevation results for the three limiters – minmod, Superbee and vanLeer – with a fine grid solution from the commonly-used ADCIRC model (Δx = 39.0625m) and verifies the phase characteristics shown in Figure 3.15. (The velocity results are similar.) The simulation parameters are the same as in Table 2.4, and the period of the open-ocean boundary condition was chosen such that the simulated wavelength satisfied $L_n/\Delta x$ equal to 30. The amplitude of the forcing function was set equal to 1.0 m. The period of the boundary condition is calculated as

$$\mathrm{T} = L_n/c = 30\Delta x \Big/ \sqrt{gh - \left(0.5\tau\frac{30\Delta x}{2\pi}\right)^2} \quad , \tag{3.53}$$

which gives T = 3029.79 seconds for the parameters given in Table 2.4. Additionally, the total domain length was set equal to 100,000m to avoid the waves reflecting at the land boundary during the simulation.

Figure 3.16b shows an expanded view of the first peak after three full periods of simulation time (10000 seconds). The first peak was chosen since it has traveled approximately one wavelength through the domain, which should result in the best comparison with the predicted propagation results, as they were calculated on a per-wavelength basis; and the long simulation time was to allow the domain to spin up from a cold start before analyzing the results. Notice that the vanLeer limiter is extremely dissipative and that the wave shape is skewed significantly. Although the wave travels at the correct speed, as evidenced by the location of the roots, the use of the peak to find the

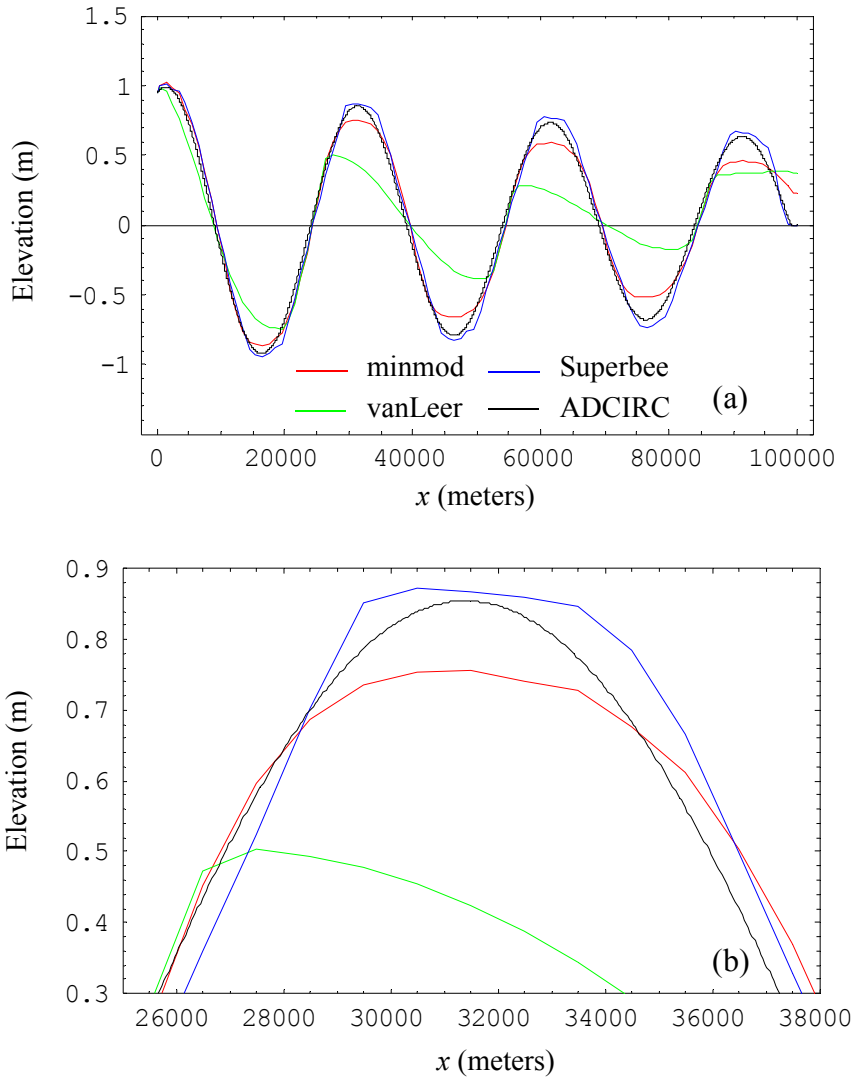


Figure 3.16 Comparison of higher-order FVM simulation results with a fine grid ADCIRC simulation after three full periods of a $30\Delta x$ wave: (a) full domain, (b) zoomed view of first peak.

phase error results in a significant lag due to this skew. It is beyond the scope of this study, however, it would be interesting to modify the "peak tracking" propagation routines presented in Chapter 2 to calculate the phase error based on the location of the roots.

Table 3.5 compares the predicted propagation characteristics, taken from the phase plots in Figure 3.15 at $L_n/\Delta x = 30$, and the measured propagation characteristics (comparing

Table 3.5 Comparison of predicted and measured propagation characteristics for higher-order FVM simulations using minmod, Superbee and vanLeer limiters.

Relative to ADCIRC	Phase error in degrees (+ lead, – lag)		Damping ratio	
	Measured	Predicted	Measured	Predicted
minmod	+1.125	–0.025	0.8841	0.8487
Superbee	–10.875	–6.025	1.0223	1.0168
vanLeer	–46.875	–48.025	0.5894	0.5986

peak location and magnitude – "peak tracking"), taken from the observed simulation behavior in Figure 3.16b. Notice that, despite the relative coarseness of the higher-order FVM results compared to the fine grid solution, the actual errors seen in the simulation are predicted by the numerical propagation tools.

3.6.2 Higher-order discontinuous Galerkin method

Similar to the higher-order FVM algorithm, the higher-order DG method cannot be analyzed analytically. Herein unlimited, where the computed slope degrees of freedom are not limited, and minmod limited DG approximations with piecewise linears are examined. The propagation curves, as generated by the numerical techniques, are presented in Figure 3.17; notice that the phase and magnitude data are discrete since there are no analytical results to present, but that the continuum dispersion behavior has been plotted for reference.

The unlimited DG with piecewise linears exhibits slight phase errors for the small wavelengths (on the order of $\pm 10°$), which are damped out, while the long physical waves have zero phase error and no damping. Similarly, the minmod limited DG using piecewise linears starts with a phase lag of $180°$ for the $2\Delta x$ wave, which is damped out, and it also exhibits perfect phase behavior for the physical wavelengths. However, the minmod limited version is slightly more dissipative in the lower wavelengths and has larger phase errors, as compared to the unlimited version.

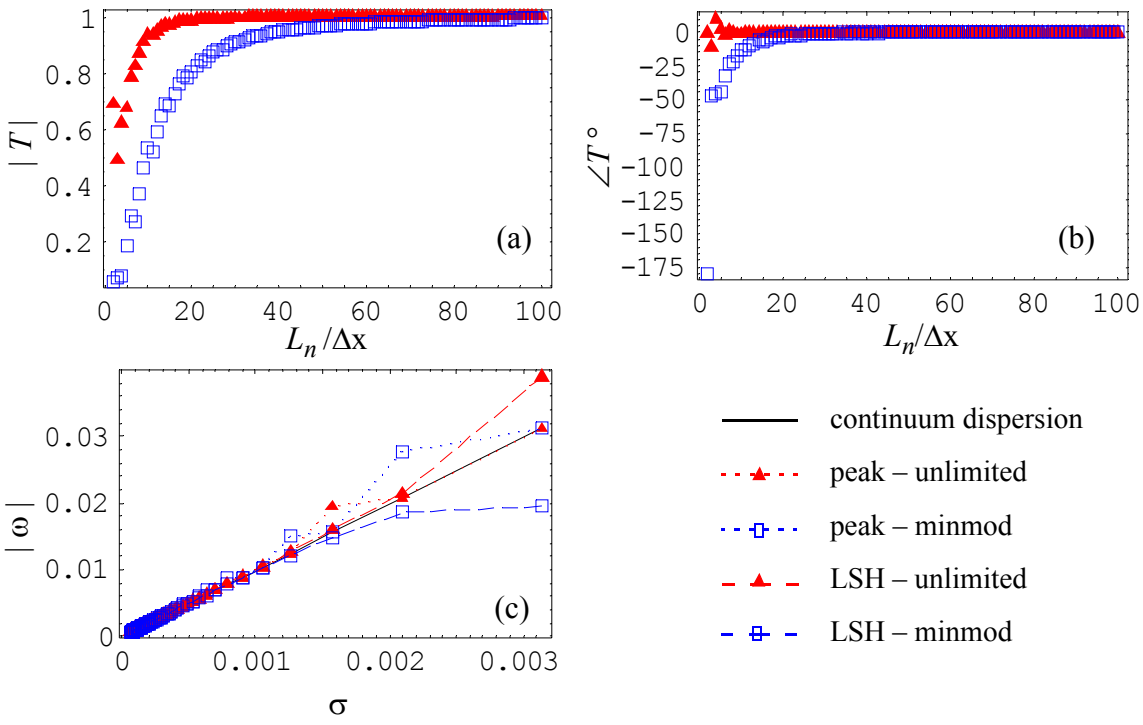


Figure 3.17 Propagation characteristics of higher-order DG methods using piecewise linears with and without slope limiters.

The dispersion curves are plotted for both the peak and least squares harmonic wave tracking techniques for comparison. The peak tracking technique results in a dispersion curve that overlies the continuum dispersion for all but the 4Δx wavelength for the unlimited version, while the minmod limited version differs at the 3Δx and 5Δx wavelengths. Meanwhile, the LSH tracking technique also results in monotonic dispersion curves, but they do not match the continuum as well.

The DG algorithm is able to produce accurate and stable results without any limiting procedure, in contrast with the FVM algorithm, which requires a limiter in the post-processing step in order to remain stable. A comparison of Figures 3.15 and 3.17, seems to indicate that the unlimited DG has very similar propagation characteristics to the Superbee limited FVM; namely, slight phase leads in the small wavelengths and minimal

damping.

The same test case that was used to verify the propagation behavior for the higher-order FVM in §3.6.1 was also used to examine the predictive capabilities of the higher-order DG propagation results. A $30\Delta x$ wavelength was simulated using the unlimited and minmod limited DG algorithm, and the first peak was compared to the fine-grid ADCIRC simulation in order to measure the propagation characteristics. Figure 3.18 shows the higher-order DG simulation results after three full periods of simulation time (10000 seconds); additionally, the low-order DG results have been plotted for reference. Notice that the unlimited higher-order DG algorithms are more accurate and that the minmod limiter in the DG framework behaves similarly to the minmod limited higher-order FVM results given above in Figure 3.16.

Table 3.6 compares the predicted propagation characteristics, taken from the phase plots in Figure 3.17 at $L_n/\Delta x = 30$, and the measured propagation characteristics (calculated by comparing the peaks), taken from the observed simulation behavior in Figure 3.18b. Notice that the damping behavior seen in the simulations is predicted quite closely by the numerical propagation tools. However, due to the relative coarseness of the DG results compared to the fine grid solution, the measured phase error for the minmod limited version does not match the predicted value. Examination of the simulation results in Figure 3.18b indicates that the minmod peak is wide and flat, such that finer resolution would result in a more accurate location for the peak.

Table 3.6 Comparison of predicted and measured propagation characteristics for higher-order DG simulations with unlimited and minmod limited piecewise linears.

Relative to ADCIRC	Phase error in degrees (+ lead, – lag)		Damping ratio	
	Measured	Predicted	Measured	Predicted
minmod	–10.875	–0.025	0.8208	0.8471
unlimited	+1.125	–0.025	1.0112	1.0013

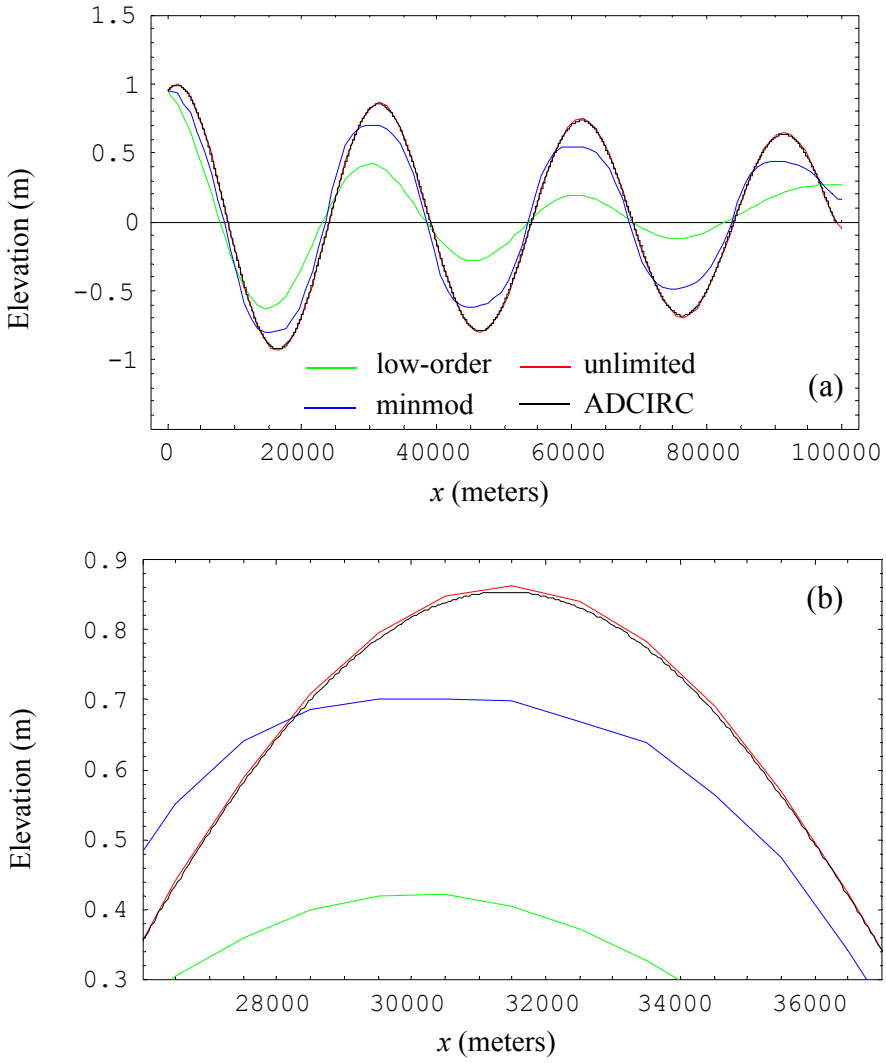


Figure 3.18 Comparison of higher-order DG simulation results with a fine grid ADCIRC simulation after three full periods of a $30\Delta x$ wave: (a) full domain, (b) zoomed view of first peak.

3.6.3 Selective lumping finite element

The numerical propagation tools can also be used to study the effect of the lumping parameter in the selective lumping finite element method. After observing the analytical Fourier behavior of the SLFE discretization scheme, as characterized by the equations in Table 2.2, it was discovered that the second lumping parameter, f, does not significantly

alter the solution. (Recall that f is the lumping parameter for the bottom friction term in Equation (1.6).) Values of bottom friction ranging from 10^{-5} to 10^{-3} sec^{-1} were examined, while e was held constant and f was allowed to vary from 0 (unlumped) to 1 (fully lumped). There was no significant change in phase or amplitude behavior, and e and f were set equal for all further studies. (Recall that the dispersion behavior of the SLFE discretization scheme cannot be studied analytically due to the split-level time marching algorithm, thus an initial study of the analytical propagation behavior was done using Fourier analysis before proceeding with numerical dispersion analysis.)

Figure 3.19 compares the numerical Fourier propagation results to the analytical results from Figure 3.14 for a fixed lumping parameter of $e = 0.9$ and a range of time steps,

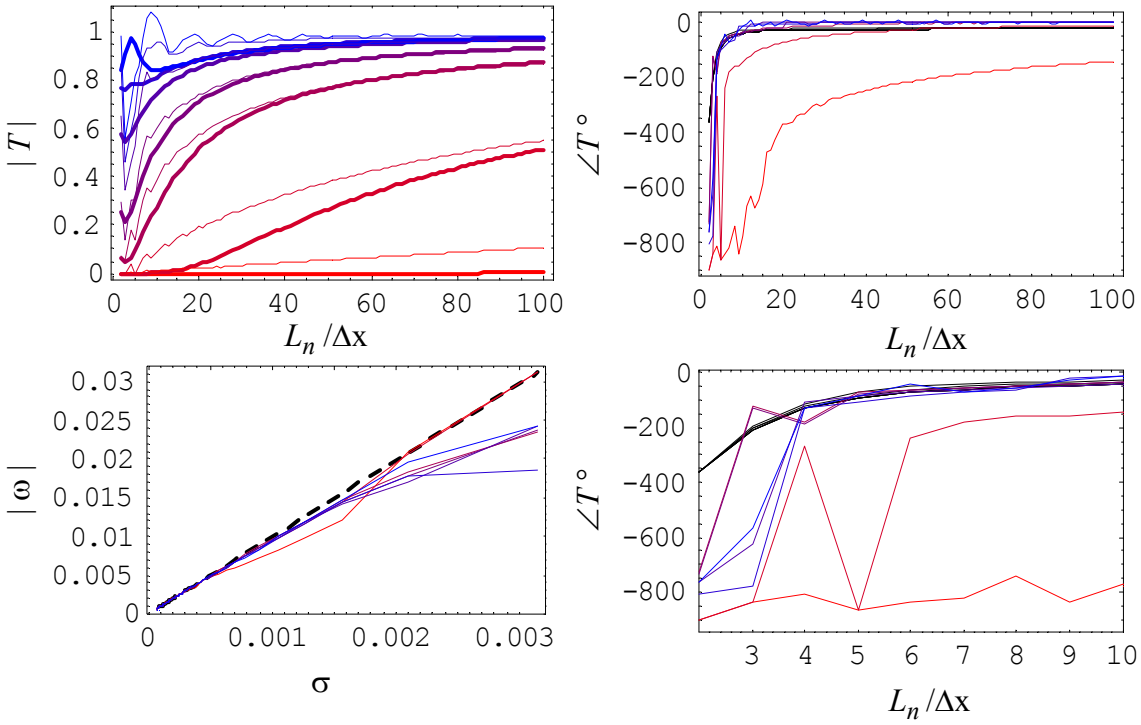


Figure 3.19 Numerically generated propagation characteristics for the SLFE with $e = f = 0.9$ and $\Delta t = [0.1 \text{ to } 75]$ seconds compared to the analytical propagation results.

Δt = [0.1, 1, 5, 10, 25, 50, 75] seconds. A zoomed view of the phase errors for the 2 to 10Δx wavelength range and numerical dispersion results are also shown. The analytical damping results are shown with the thick curves, while the numerical damping results are shown with thin curves. Meanwhile, the analytical phase results are all shown with black curves since they do not differ significantly and the numerical phase results are shown in color. For all colored plots, the curves go from small Δt values in red to large Δt values in blue.

Notice first that the numerically derived damping results for the two smallest time steps (Δt = 0.1 and 1.0 seconds) and the two largest time steps (Δt = 50 and 75 seconds) are slightly better than those derived analytically. Notice also that the phase errors for the largest time steps, which were derived from the numerical propagation analysis, approach an asymptotic value of 0.0 while the analytical results only reach a value of –25 degrees. Also, the analytical phase errors for all of the time step sizes go to –360 degrees at the 2Δx wavelength; but the numerically generated phase errors start between –720 and –900 degrees. This indicates that the 2Δx wave is not remaining stationary, as predicted analytically, but is actually propagating in the reverse direction. Despite these large phase errors, generally enough damping is provided by each of the simulated time steps to either remove or severely damp these spurious wavelengths. However, only the extremely dissipative nature of the model using a time step of 0.1 seconds is able to completely remove the 2Δx noise, as indicated by the red dispersion relationship. Notice that all of the other time steps have solutions that remain monotonic, however they do fold away from the continuum dispersion relationship to some extent.

Figure 3.20 presents the numerical propagation results over the entire range of lumping coefficients, unlumped (e, f = 0.0) to fully lumped (e, f = 1.0). The maximum stable time step was used for each lumping value and is summarized in Table 3.7; all other parameters are as in Table 3.4.

Notice that larger time steps can be taken when the algorithm is unlumped, however

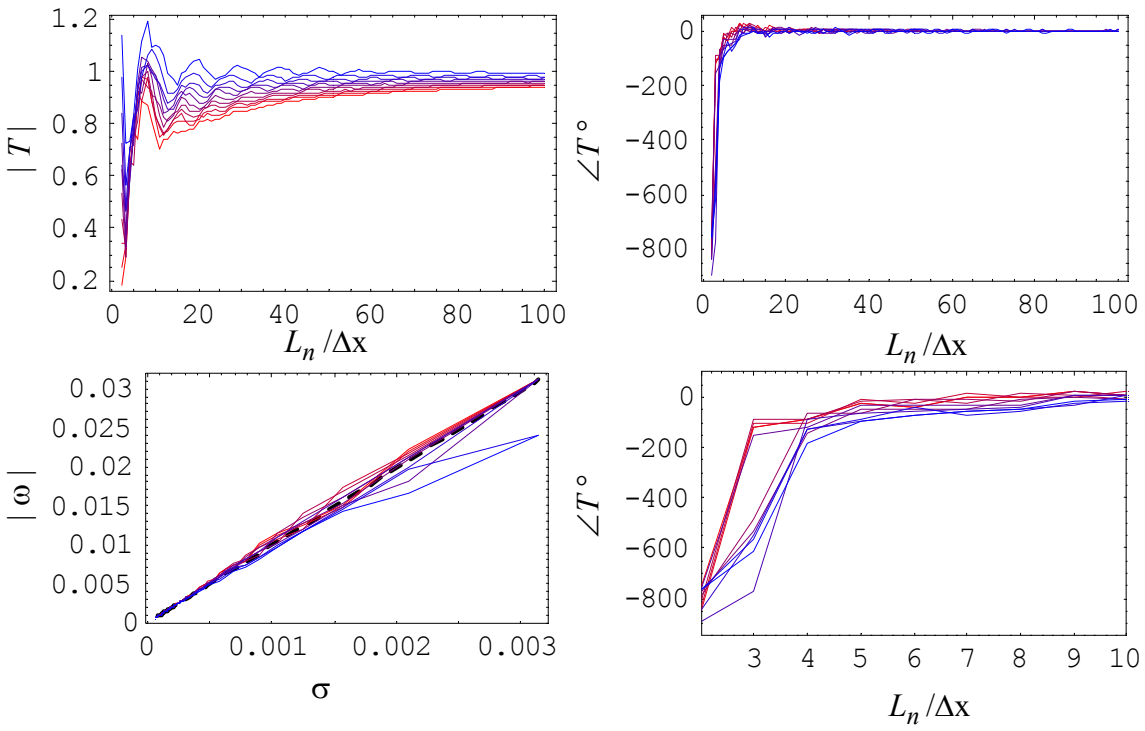


Figure 3.20 Numerically generated propagation characteristics for the SLFE over the range of lumping parameters, $e = f =$ [0.0 to 1.0], using the maximum stable time step for each lumping value.

Table 3.7 Maximum stable time steps for each value of the lumping parameter.

e, f	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
max Δt (sec)	135	135	130	125	120	115	110	100	90	75	45

the model is overly dissipative for the smaller lumping coefficient values. Notice also, that the phase errors for all values of the lumping coefficient start out around –800 degrees, indicating, as in the previous example, that the $2\Delta x$ wave is propagating in the opposite direction. All of the values produce monotonic dispersion relations, however as the algorithm nears being fully lumped ($e, f = 0.9$ and 1.0) the dispersion relation begins to fold away from the continuum. Although all of the other lumping values are near the continuum

relation, the curves for $e, f = 0.5$ and 0.7 are an exact match, which may indicate nothing other than the time steps used for these simulations were closest to the maximum stable. (The maximum time steps were chosen by increasing the time step in 5 second increments until instability was reached and then using the previous value.)

Figures 3.19 and 3.20 demonstrate that the SLFE algorithm is extremely dependent on both the lumping parameter and the time step. In general, each of the values for the lumping parameter would behave similarly as the time step is decreased from the maximum stable value. Therefore, for the validation of the propagation behavior, the lumping coefficient was set to 0.9. The same test case that was used to validate the propagation behavior for the higher-order FVM in §3.6.1 was also used to examine the predictive capabilities of the SLFE propagation results. A $30\Delta x$ wavelength was simulated using three of the test time steps Δt = [0.1, 10.0 and 50.0] and the same grid spacing of Δx = 1000m, and the first peak was compared to the fine-grid ADCIRC simulation in order to measure the propagation characteristics. Figure 3.21 shows the SLFE simulation results after three full periods of simulation time (10,000 seconds). Notice that the elevation results simulated using the smallest time step are utter nonsense – the entire solution is essentially damped out. Obviously, small time steps are not recommended with this algorithm.

Table 3.8 compares the predicted propagation characteristics, taken from the phase plots in Figure 3.19 at $L_n/\Delta x = 30$, and the measured propagation characteristics (calculated by comparing the peaks), taken from the observed simulation behavior in Figure 3.21b. No

Table 3.8 Comparison of predicted and measured propagation characteristics for SLFE simulations with the lumping coefficient set to 0.9.

Relative to ADCIRC	Phase error in degrees (+ lead, – lag)		Damping ratio	
Δt (sec)	Measured	Predicted	Measured	Predicted
0.1	—	– 294.0	—	0.0319
10.0	– 4.875	– 0.025	0.8957	0.8067
50.0	– 4.875	– 2.401	0.9830	0.9593

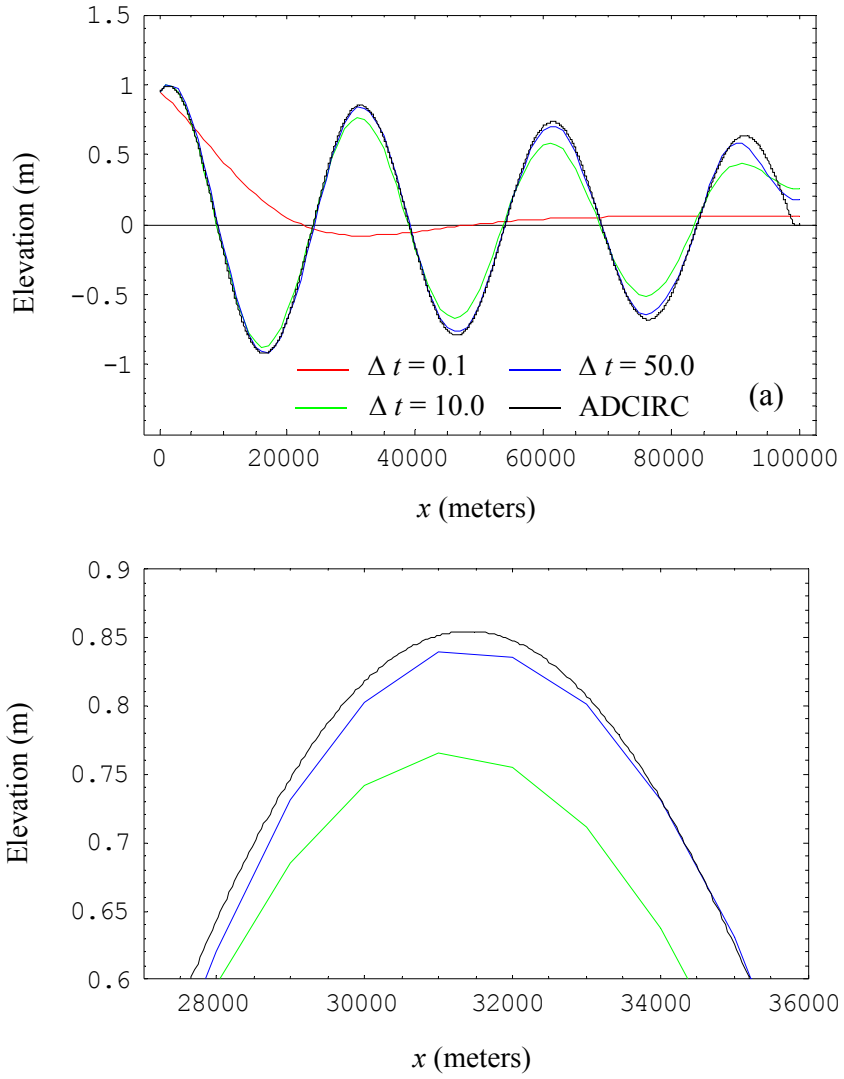


Figure 3.21 Comparison of SLFE simulation results using a lumping parameter of 0.9 with a fine grid ADCIRC simulation after three full periods of a 30Δ*x* wave: (a) full domain, (b) zoomed view of first peak.

measured propagation results are calculated for the smallest time step, since the results are completely damped out. Notice that, despite the relative coarseness of the SLFE results compared to the fine grid solution, the actual errors seen in the simulation are predicted quite closely by the numerical propagation tools.

3.7 Summary of model characteristics

Table 3.9 summarizes the spatial and temporal accuracy of each of the algorithms. For those algorithms that were analyzed above in §3.3 using truncation error analysis, the grid convergence results from §3.4 are also provided for comparison. See the referenced sections for more details about these results.

Table 3.9 Summary of temporal and spatial accuracy studies for the study algorithms.

	Temporal accuracy		Spatial accuracy		
	analytical	numerical[b]	analytical	best-fit [c]	peak [c]
SFD	2 (2) [a]	1.95 (1.94)	2 (2)	0.99 (0.97)	1.00 (1.21)
PLF	2 (1)	1.15 (1.13)	2 (2)	1.46 (1.01)	1.83 (1.13)
ADCIRC (GWC)	2 (2)	1.96 (2.00)	2 (2)	1.44 (1.02)	1.76 (1.65)
SLFE	1 (1)	-0.68 (-0.99) [d]	2 (2)	0.96 (0.90)	0.99 (0.96)
low-order FVM / DG	1 (1)	1.00 (1.00)	1 (1)	0.99 (0.95)	1.00 (0.98)
FVM (minmod)	na	1.08 (1.07)	na	0.96 (1.06)	1.00 (1.18)
FVM (Superbee)	na	1.28 (1.27)	na	1.02 (1.25)	1.07 (1.33)
FVM (vanLeer)	na	1.00 (1.00)	na	0.95 (0.94)	0.99 (0.98)
DG (unlimited)	na	1.07 (1.07)	na	1.51 (1.66)	1.86 (1.80)
DG (minmod)	na	0.99 (0.99)	na	1.01 (1.04)	1.04 (1.21)

[a] The first number indicates the accuracy of the continuity approximation and the number in parentheses indicates the accuracy of the momentum approximation.

[b] The temporal numerical accuracy is calculated by averaging the converged accuracies from the L_2 and L_∞ error norms for each state variable.

[c] The spatial numerical accuracy is calculated for both the average of the best-fit lines and the peak rates for the L_2 and L_∞ error norms.

[d] The slope of the temporal convergence plots for the SLFE is negative because the quality of the solution decreases as the time step is resolved away from the maximum stable value.

Notice that the numerically generated spatial convergence results for the SFD and SLFE algorithms do not live up to the derived second-order rates from truncation error analysis, while the PLF and ADCIRC best-fit convergence rates are also not exactly second-order but are closer (the peak convergence rates for the PLF and ADCIRC models are closer to 2). The lower convergence rate for the SLFE algorithm is likely due to the

interplay between the temporal and spatial refinement for the SLFE algorithm; recall that a different (max stable) time step was used for each grid spacing. Additionally, the numerically generated temporal convergence results for the PLF algorithm do not live up to the derived second-order rate for the continuity equation. Most importantly, notice that the spatial convergence rates for the unlimited higher-order DG algorithm are greater than those for the other higher-order discontinuous algorithms (FVM and DG) and are comparable to the ADCIRC model.

Table 3.10 summarizes the stability limits for each of the algorithms, using the parameters given in Table 3.4. Numerically, these limits were determined by running the algorithms with successively larger time steps (to the nearest second) until the solution became unstable. For those algorithms that were analyzed above in §3.5.3 using the propagation factors, the analytical results are also provided for comparison. Notice that the numerical results agree quite well with the derived limits. Also, note that due to the dissipative nature, the FVM using piecewise linears and either the minmod or vanLeer limiters is more stable than the FVM using the Superbee limiter; although the quality of the solution is decreased at these high time steps. Additionally, the FVM algorithm using vanLeer limiting has a comparable stability limit to the low-order FVM, which is expected

Table 3.10 Analytical and numerical stability limits for the study algorithms.

maximum stable Δt (sec)	analytical	numerical
SFD	unconditional	unconditional
PLF	58.12	55
ADCIRC (GWC)	109.08	110
SLFE	79.23	65
low-order FVM / DG	100.46	100
FVM (minmod)	na	30
FVM (Superbee)	na	7
FVM (vanLeer)	na	90
high-order DG (unlimited)	na	3
high-order DG (minmod)	na	30

since they are both overly dissipative and have similar damping behavior (refer to Figure 3.15). Finally, notice that the stability limit for the minmod limited DG algorithm is the same as that for the minmod limited FVM, and is significantly higher than that for the unlimited DG.

From the propagation results, all of the algorithms except for the PLF model, the low-order FVM/DG algorithms, the vanLeer limited FVM and the SLFE algorithm with small Δt or e, exhibit good phase behavior. The PLF algorithm exhibits a folded dispersion curve, indicating that spurious oscillations can hinder the solution; the low-order FVM/DG algorithm is overly-dissipative and damps out the physical waves; the vanLeer limited FVM has phase errors in the physical wavelengths; and the SLFE with low Δt or e is overly-dissipative. Of the discontinuous algorithms, the low-order FVM/DG method is overly dissipative in the long-wave physical range; and a higher-order approximation is needed to correct this. The vanLeer limiter for the FVM does not appear to be a good choice, as it is also overly dissipative and has a phase lag in the physical wavelengths; but any of the other limiting choices for the FVM or the unlimited and minmod limited DG methods have good propagation characteristics.

Additionally, a simple test case in 1D with M_2 tidal forcing (period = 44,712 seconds, amplitude = 1m) at the ocean boundary is used to compare the CPU times for each of the study algorithms. For these tests, a Dell Dimension 2400 with a 2.66GHz Pentium® 4 processor and the Lahey-Fujitsu® Fortran compiler for Windows was used. Each of the algorithms is used in turn to simulate the tidal response for just over three full periods (135,000 seconds). The maximum stable time step for each algorithm and a grid spacing of 640 elements is used for these simulations, while all other parameters are as in Table 3.4. The maximum stable time step and resulting CPU times for each algorithm are summarized below in Table 3.11.

Remarkably, the SFD algorithm is capable of using a time step as large as 1000.0

Table 3.11 CPU times for three periods of M_2 tidal forcing.

Model	max stable Δt (sec)	CPU usage (sec)
SFD	1000.0	0.0156
PLF	2.0	7.6094
ADCIRC (GWC)	10.0	1.5156
SLFE (e=0.9)	2.0	8.9688
low-order FVM / DG	5.0	1.0781
higher-order FVM (minmod)	0.5	17.672
higher-order FVM (Superbee)	0.1	96.313
higher-order FVM (vanLeer)	5.0	2.1406
higher-order DG (unlimited)	0.05	132.64
higher-order DG (minmod)	0.5	35.953

seconds before the quality of the solution begins to degrade. Even larger time steps are possible since the algorithm is unconditionally stable, however the quality of the solution declines rapidly past this point. The ADCIRC model requires 1/100th of this time step and is thus 100 times slower than the SFD model. However, the more flexible finite element grid scheme of the ADCIRC model makes it more attractive for actual applications, which most often have rather complicated domains.

The PLF and SLFE algorithms have the same stability limit for this grid spacing, one-fifth of the allowable time for the ADCIRC model. Since the PLF algorithm uses the same tridiagonal solver as the ADCIRC model, it scales accordingly and requires approximately five times the CPU usage of the ADCIRC model. However, although the SLFE algorithm is explicit in nature, such that no matrix solutions are required, it is a two-step procedure and thus takes slightly more CPU time than the PLF algorithm.

Meanwhile, the low-order FVM/DG algorithm (recall that the FVM and DG algorithms are equivalent when piecewise constant approximating spaces are used) requires one-half of the time step allowed by the ADCIRC model; however, it uses slightly less CPU time since the algorithm is explicit. The addition of higher-order approximations

and the vanLeer slope limiter does not significantly increase the CPU usage; however, as seen in §3.4, no spatial or temporal accuracy is gained (relative to the low-order FVM) by using this model. The addition of higher-order approximations and the minmod limiter requires a time step equal to one-tenth of that allowed with the low-order version; however, the CPU usage is approximately sixteen times as large due to the extra computations required for the limiter. Use of the Superbee limiter instead, requires a time step one-fiftieth of that allowed with the low-order version; but requires nearly one hundred times as much CPU usage. Not only is the superbee limiter more limited by stability constraints, but it also requires more computations in the limiting process itself. From these (and other) results with the FVM algorithm, it was decided that only the minmod limiter would be examined for the DG algorithms.

Finally, the unlimited higher-order DG algorithm requires a time step two hundred times smaller than that allowed by the ADCIRC model and uses about 90 times as much CPU time. Meanwhile, the minmod limited version requires a time step 20 times smaller than the ADCIRC model and uses about 24 times as much CPU time. Although the minmod limited version is more computationally intensive than the unlimited version, the ability to take larger time steps ultimately makes it worthwhile. The higher-order DG algorithms are clearly more expensive than the ADCIRC model; although they do provide slightly more spatial accuracy, as seen in Figure 3.6 in §3.4.

3.8 Conclusions

Herein, the discontinuous class of algorithms has been compared to the more traditional continuous algorithms for the solution of the shallow water equations and found to be viable alternatives. The propagation characteristics of the various FVM and DG implementations are comparable to the popular GWC approach. The stability limits of the low-order FVM/DG and higher-order FVM with a vanLeer limiter are comparable to the ADCIRC algorithm. Although the DG class has more restrictive stability limits, from an

implementation point of view, it is more flexible than the FVM, since it has a finite element basis and naturally incorporates higher-orders. Additionally, the DG exhibits higher spatial convergence rates and has comparable rates to the ADCIRC model. However, due to the more restrictive stability limits, and thus higher CPU times, the DG class of algorithms is not as computationally efficient as the ADCIRC model. In two- or three-dimensional applications the number of finite element edges, where much of the extra computational effort arises in the discontinuous algorithms, can grow quickly, resulting in extremely long computational times. Thus it might be beneficial to use the DG algorithm in a limited domain and use the ADCIRC model in the remainder of the domain. Analysis of various techniques for coupling the DG and ADCIRC models is the topic of the next chapter.

CHAPTER 4

Coupled Continuous and Discontinuous Finite Element Methods for the Shallow Water Equations

4.1 Introduction

Shallow water equations (SWE) are used to model hydrodynamics in lakes, estuaries, coastal regions, and other impoundments. The system of 2D SWE arises from the depth-averaged Navier-Stokes equations and consists of a continuity equation with either a conservative or non-conservative form of the momentum equation. Numerical techniques are required to solve the SWE over complex domains, and it is desirable that these algorithms produce accurate results without introducing numerical noise (spurious oscillations) into the solution. In this study, the linearized 1D versions of the SWE will be used in order to make use of the propagation tools in Chapter 2. For convenience, these equations, as well as the linear form of the generalized wave continuity equation (GWC), will be repeated here, although they are also presented in §1.2.2. The linear form of the primitive continuity equation is

$$\zeta_t + h u_x = 0 \qquad \textbf{(4.1)}$$

and the inviscid linear momentum equation is

$$u_t + \tau u + g\zeta_x = 0\,. \qquad \textbf{(4.2)}$$

The linear form of the generalized wave continuity equation is

$$\zeta_{tt} + G\zeta_t + (G-\tau)hu_x - gh\zeta_{xx} = 0 \quad \textbf{(4.3)}$$

where ζ is the deviation of the surface elevation from the datum, h is the bathymetry, u is the velocity in the x coordinate, τ is the linearized bottom friction parameter, g is the gravitational constant (9.81 m^2/s), and G is the generalized wave continuity weighting parameter. When G approaches infinity, the GWC equation tends toward primitive continuity, while for small G, it tends toward wave continuity. Note that the non-conservative and conservative forms of momentum are identical when the equations are linearized. In the remainder of this chapter, Equations (4.1) and (4.2) together will be referred to as the primitive SWE, while Equations (4.3) and (4.2) together will be referred to as the ADCIRC model.

The wide range of applications used in shallow water modeling necessitate robust and computationally-efficient algorithms that maintain a high degree of accuracy. For example, storm surge predictions require immediate results with a certain degree of accuracy, while coastal dredging applications require accuracy but not immediate results. Additionally, the large and complex simulation domains that are used in practice often have widely varying bathymetry; therefore, flow regimes from highly advective in the shallow regions to more diffusive in the deeper regions are encountered within a single simulation. With such diverse flow regimes coupled with large domains for practical applications, it is reasonable to assume that a single algorithm may not be able to adequately model the varied physical processes, while maintaining accuracy and computational efficiency.

In particular, highly advective regions near the shorelines may result in mass balance errors in the existing GWC algorithm, which is employed in the production code, ADCIRC (ADvanced 3D CIRCulation model *Luettich et al.* [1992, 2003]). Historically, the GWC algorithm was developed for slowly evolving fields in deeper continental shelf and oceanic regions; it is not as well-suited for highly advective flows. On the other hand,

discontinuous Galerkin (DG) algorithms, which are relatively new in the surface water literature, are capable of handling such advective flows with sharp fronts and can use the same triangular meshing scheme (in 2D/3D models) as the ADCIRC model.

DG finite element methods are similar to traditional Galerkin FE methods in that both solve the weak form of the equations by multiplying by a weight function and integrating. As in traditional Galerkin approximations, the weight functions are identical to the interpolating functions. There are, however, two crucial differences between traditional continuous Galerkin (CG) approximations and discontinuous approximations. The first distinction is that DG methods allow discontinuities at the element interfaces, which in effect disconnects adjacent elements and allows each to be solved "independently" from the others. (As will be seen, each element does require information from adjoining elements, but this information enters in explicitly for the explicit time marching algorithms used in practice and does not couple the elements.) The second distinction follows from the first in that DG methods integrate over a single element using local weight and interpolating functions, while continuous methods integrate over the entire domain and use global weight functions, which are zero outside of the element of interest. These two differences provide certain advantages to the DG method including: local *p*-refinement (i.e., each element has its own basis and weight function allowing higher order interpolates in known problem areas), local *h*-refinement (i.e., each element is not required to match up with its neighbors and hanging nodes do not pose a problem), the ability to capture discontinuous solutions and handle shock fronts, and local mass conservation since equations are satisfied on the element level rather than globally.

Experience with the finite volume code (which, as discussed in Chapter 3, is equivalent to DG for low-order interpolation), UTBEST, which is based on the work of *Chippada et al.* [1998], indicates that such shock-capturing algorithms are more computationally-intensive than the GWC algorithm. Also in deep water regions, they are no more accurate than the GWC. Thus, there is no "best" algorithm. Rather, an optimal

solution may be to use the computationally-efficient GWC in deep water and switch to the DG in shallow, near-shore regions. However this raises two new research issues: how to couple the two algorithms and where in the domain to switch from one algorithm to the other. Note that the coupling is further complicated by the fact that in the GWC algorithm, elevations and velocities are calculated at the node points, while in the DG algorithm, these quantities are element-centered.

Dawson and Proft have begun examining the first issue, how to couple the two algorithms. In their first paper [2002b], they examine two approaches for coupling continuous and discontinuous algorithms. The first approach, denoted by DG/CG, is to discretize the primitive continuity equation with a local discontinuous Galerkin (LDG) scheme and the momentum equation with standard continuous Galerkin (CG) finite elements. In the second approach, denoted by DG/NIPG, they discretize the continuity equation with the LDG scheme and the momentum equation with a particular DG scheme for second-order problems, the nonsymmetric interior penalty Galerkin (NIPG) method of *Rivière et al.* [1999, 2001]. They derive *a priori* error estimates and find an optimal convergence rate of h^2, using linear interpolants, for both schemes. In a second paper, *Dawson and Proft* [2004] extend the first equation coupling scheme (DG/CG) and further couple that system to a domain-wide CG discretization, as is done in the GWC algorithm; thus they have coupled DG and CG algorithms through both equation and subdomain coupling. In one subdomain the DG/CG algorithm is applied, and in the other subdomain the ADCIRC model is employed; coupling at the subdomain interface is accomplished through the boundary fluxes. Although the momentum equation is discretized with CG elements in each subdomain, the velocity solution at the subdomain interface is allowed to be discontinuous. This work is explained in further detail by *Proft* [2002] in her Ph.D. dissertation.

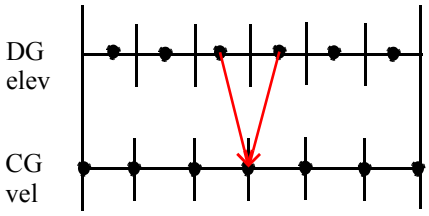
The study herein seeks to further analyze the first question, how to couple the two algorithms, and answer the second question, where in the domain to switch from one

algorithm to the other, by studying the various coupling techniques in a 1D framework and using the numerical propagation analysis tools developed in Chapter 2. Additionally, the distortion and phasing errors at the interface are examined in the context of the wave propagation characteristics.

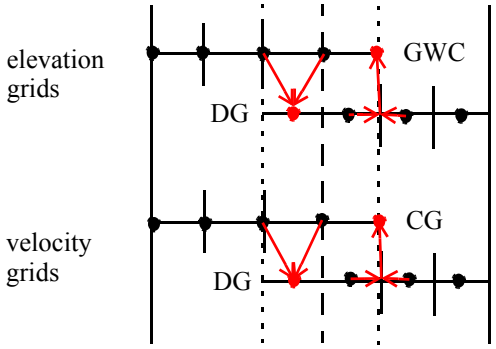
4.2 Notation and coupling method descriptions

In this study, two types of coupling will be examined. The term "equation coupling" will be used to refer to coupling techniques where the continuity equation is discretized using one algorithm and the momentum equation is discretized with another, as in the DG/CG and DG/NIPG schemes of *Dawson and Proft* [2002b]. The term "subdomain coupling" will be used to refer to coupling schemes where the simulation domain is broken into subdomains with different algorithms used to obtain the solution in each subdomain, as in *Dawson and Proft* [2004]. Also, equation coupling will be noted with a slash, where the left-hand-side of the slash denotes the discretization for the continuity equation and the right-hand-side denotes the discretization for the momentum equation, e.g. DG/CG. Subdomain coupling will be noted with a double arrow to indicate that information must be passed between subdomains, e.g. $\mathrm{DG} \leftrightarrow \mathrm{ADCIRC}$.

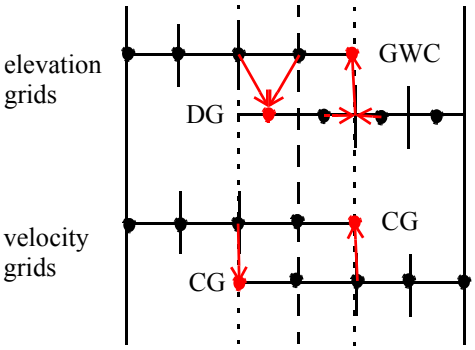
This study will examine three degrees of coupling for the solution of the shallow water equations in one spatial dimension. In order of complexity these are: Method 1) DG/CG - equation coupling of DG continuity and CG momentum in a single domain, Method 2) $\mathrm{DG} \leftrightarrow \mathrm{ADCIRC}$ - subdomain coupling of domain-wide DG to ADCIRC formulation, and Method 3) $\mathrm{DG/CG} \leftrightarrow \mathrm{ADCIRC}$ - subdomain coupling of DG/CG equation coupled scheme to ADCIRC formulation. For the subdomain coupling (Methods 2 and 3), an overlapping element scheme is used to couple the subdomains to one another, which is typical of parallel domain decomposition algorithms. Schematic diagrams of these coupling schemes in 1D are shown below in Figure 4.1, where the subdomain interface is indicated by a dashed line and overlapping elements by a dotted line for Methods 2 and 3.



Method 1) Requires interpolation (shown by arrows) to translate DG elevations into CG elevations before computing the CG velocities.



Method 2) Computes average of nodal values to pass from CG to DG and average of DG interface discontinuities to pass from DG to CG (passing indicated by arrows) at the overlapping elements.



Method 3) Same passing as in Method 2 for the elevation grids and direct passing of nodal values for the velocity grids, as both are CG grids. Note that the velocity solution at the interface can be discontinuous since a value will be calculated for it within each subdomain. Interpolation is required within the DG/CG subdomain, as in Method 1. Overlapping elements for passing between subdomains.

Figure 4.1 Schematic diagrams of coupling methods. Dashed vertical lines indicate subdomain interface and dotted vertical lines indicate overlapping elements. Red arrows and dots indicate shared information.

It was hoped that boundary flux passing, as in *Dawson and Proft* [2004], could also be examined as a method for coupling subdomains, but this was not possible. In order to use the phase analysis tools that were developed in Chapter 2, it is necessary to use the linear form of the shallow water equations. However, the linear form does not lend itself to flux coupling because all of the total-flux terms ($Q=Hu$) are removed in the linearization. While it would have been instructive to examine the phase properties of the flux coupling scheme, *Dawson and Proft* [2004] have already shown that the solution passes through the coupling interface without distortion. Thus, this flux coupling technique will not be

examined in this study; rather an alternative coupling with overlapping nodes and "natural" treatment of boundary information is used. (Within the DG and DG/CG subdomains, the algorithm will essentially be unaware of the subdomain interface, since computations will begin at an interior element.) Each of the three methods will be derived in the following sections.

Throughout the derivations in the subsequent sections, inner product notation $(\ , \ )_{\Omega_e}$ will denote integration over an element and the notation $\langle \ , \ \rangle_{\Omega_e}$ will denote integration over the element boundary, which in one dimension is just a point evaluation.

4.3 Method 1: DG/CG equation coupling

For this coupling method, the primitive continuity equation is discretized using discontinuous Galerkin (DG) finite elements and the momentum equation is discretized using continuous Galerkin (CG) finite elements. The CG model for the momentum equation is derived using standard Galerkin FE with linear approximating functions. For the DG model of the primitive continuity, piecewise constant and piecewise linear approximating functions are examined. Grid space notation on an element is shown in Figure 4.2.

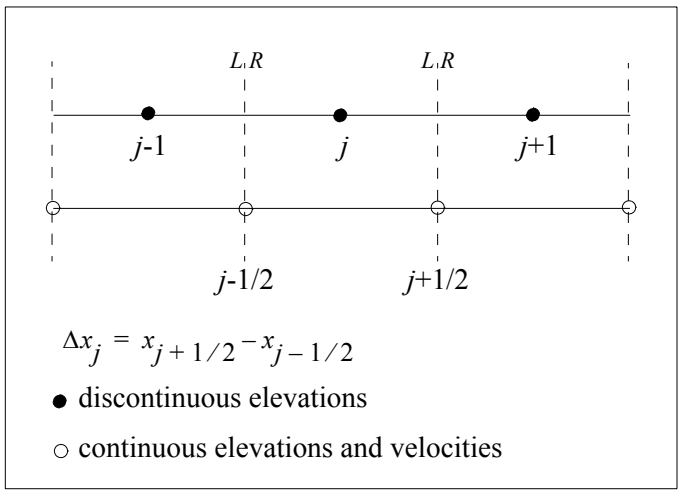


Figure 4.2 Grid space notation for DG/CG model.

The first step is to solve for the new DG elevations using the CG velocities and DG elevations from the previous time step. Then these new DG elevations are interpolated onto the CG grid space. The final step is to solve for the new CG velocities using the CG velocities from the previous time step and the CG elevations from the previous and current time steps.

4.3.1 DG solution of the primitive continuity equation

The starting point for the DG/CG discretization of the primitive SWE is the "local" weak form of the continuity equation. Multiply the continuity equation by a discontinuous weight function, $\phi_i(x)$, and then integrate by parts over an element, Ω_e, to get

$$(\frac{\partial \zeta}{\partial t},\phi_i)_{\Omega_e} - (hu,\frac{\partial \phi_i}{\partial x})_{\Omega_e} + \langle hu \cdot n_e, \phi_i^I \rangle_{\partial\Omega_e} = 0 \qquad \textbf{(4.4)}$$

where n_e is the unit outward normal at the element boundary, ϕ_i^I indicates that the weight function should be evaluated internal to the boundary. The explicit functional dependence of the variables, $u(x,t)$ and $\zeta(x,t)$, and the weight function has been left off for brevity.

As in the derivation of the domain-wide DG discretization of the SWE in §B.4 of Appendix B, Roe's linearization is used to calculate the numerical flux at the boundary. There it was shown that the appropriate flux for the continuity equation was

$$F_1 = \frac{1}{2}[h(u_L + u_R) - a(\zeta_R - \zeta_L)] \qquad \textbf{(4.5)}$$

where $a = \sqrt{gh}$ is the wave celerity, and the subscripts $_L$ and $_R$ denote the left and right states at the boundary. Notice that this approximation adds an additional term to the simple arithmetic average of the left and right fluxes, hu. This second term basically adds numerical diffusion for stabilization.

Now observe that in 1D the boundary integrals reduce to point evaluations and

replace the boundary term as follows:

$$\langle hu \cdot n_e, \phi_i^I \rangle_{\partial\Omega_e} = \frac{1}{2}[h(u_L + u_R) - a(\zeta_R - \zeta_L)] \cdot n_e \phi_i(x) \Big|_{x_{j-1/2}}^{x_{j+1/2}} \quad \textbf{(4.6)}$$

where the element notation is as in Figure 4.2 above.

The next step is to approximate the dependent variables with basis functions. Herein, both piecewise constant and piecewise linear Legendre bases are examined for the DG elevation approximations, while linear Lagrange bases are used for the CG velocity approximations (in anticipation of the CG discretization in §4.3.3). The fully discrete DG equations will be worked out individually for each of these bases in the following subsections.

Piecewise constant DG approximation

The structure of an element with piecewise constant approximations for the elevations is shown below in Figure 4.3. Note that the only unknown in each element is the elevation average.

The approximations on an element are given as

$$\zeta^h = \zeta_j \phi_0 \text{ and} \quad \textbf{(4.7)}$$

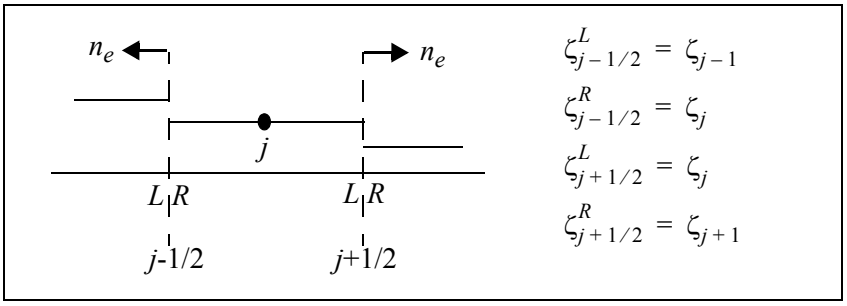


Figure 4.3 Structure of element *j* using piecewise constant basis functions for DG elevation.

$$u^h = u_{j-1/2}\omega_0 + u_{j+1/2}\omega_1 \tag{4.8}$$

where the DG basis function is simply $\phi_0 = 1.0$ and the CG basis functions (ω_0 and ω_1) are standard linear Lagrange functions.

Now substitute the approximations (4.7) and (4.8) for the state variables and the boundary integral (4.6) into Equation (4.4) to arrive at

$$\frac{\partial}{\partial t}(\zeta_j\phi_0,\phi_0)_{\Omega_e} - (h(u_{j-1/2}\omega_0 + u_{j+1/2}\omega_1),\frac{\partial\phi_0}{\partial x})_{\Omega_e} +$$

$$\frac{1}{2}[h(u_L + u_R) - a(\zeta_R - \zeta_L)] \cdot n_e\phi(x)\Big|_{x_{j-1/2}}^{x_{j+1/2}} = 0 \tag{4.9}$$

where the second integral term vanishes since the derivative of the basis function is zero, and the first integral evaluates simply to Δx_j. Also, note that since we are using continuous approximating functions for the momentum equation, the velocity at the boundary is constant such that $u_L = u_R$. With these substitutions and some simplification, the final approximation is given as

$$\frac{\partial}{\partial t}\Delta x_j\zeta_j + \frac{1}{2}[2h(u_{j+1/2} - u_{j-1/2}) - a((\zeta_{j+1} - \zeta_j) - (\zeta_j - \zeta_{j-1}))] = 0 \tag{4.10}$$

where the elevation boundary contributions are kept distinct for ease of boundary condition application.

Explicit time stepping is used for the temporal discretization, such that the fully discretized approximation for the DG elevations with piecewise constants is given as

$$\frac{\Delta x_j}{\Delta t}(\zeta_j^{k+1} - \zeta_j^k) + \frac{1}{2}[2h(u_{j+1/2}^k - u_{j-1/2}^k) - a((\zeta_{j+1}^k - \zeta_j^k) - (\zeta_j^k - \zeta_{j-1}^k))] = 0 \tag{4.11}$$

where k represents the known information from the previous time step and k+1 represents

the unknown values.

Piecewise linear DG approximation

The structure of an element with piecewise linear Legendre approximations for the elevation is shown below in Figure 4.4. Note that the variables are evaluated at the element center such that the unknown values for the interpolate in the j^{th} element are the elevation element average (indicated in the figure with a solid dot) and "slope" given at point j, which is indicated in the figure with a solid line. The elevation approximation in this basis is written as

$$\zeta^h = \zeta_j^{(0)}\phi_0 + \zeta_j^{(1)}\phi_1 \qquad \textbf{(4.12)}$$

where $\zeta_j^{(0)}$ and $\zeta_j^{(1)}$ are the elemental average and "slope" respectively. The basis functions are given as

$$\phi_0=1 \text{ and } \phi_1 = \frac{12}{\Delta x_j}(x - x_j), \qquad \textbf{(4.13)}$$

where $12/\Delta x_j$ is a normalization factor, such that the slope itself is given as

$$\text{slope} = \frac{12}{\Delta x_j}\zeta_j^{(1)}. \qquad \textbf{(4.14)}$$

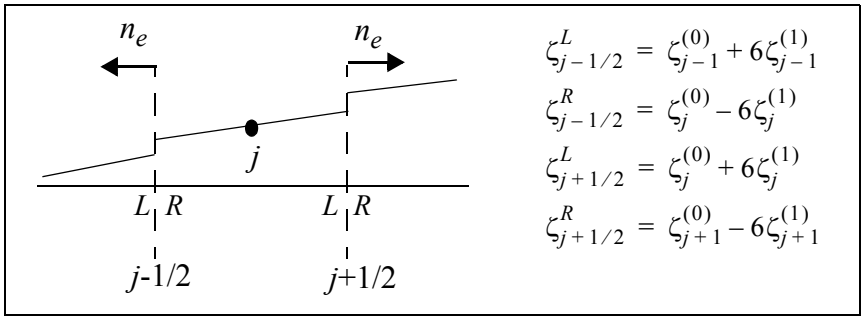


Figure 4.4 Structure of element j using piecewise linear Legendre basis functions for DG elevation.

The velocity approximation is the same as that given in (4.8).

Now substitute the approximations (4.13) and (4.8) for the state variables and the boundary integral (4.6) into Equation (4.4) to arrive at

$$\frac{\partial}{\partial t}((\zeta_j^{(0)}\phi_0 + \zeta_j^{(1)}\phi_1),\phi_i)_{\Omega_e} - h((u_{j-1/2}\omega_0 + u_{j+1/2}\omega_1),\frac{\partial \phi_i}{\partial x})_{\Omega_e} +$$

$$\frac{1}{2}[h(u_L + u_R) - a(\zeta_R - \zeta_L)] \cdot n_e \phi_i(x)\Big|_{x_{j-1/2}}^{x_{j+1/2}} = 0. \quad \textbf{(4.15)}$$

Note that Legendre bases are orthogonal, such that $(\phi_j,\phi_i)_{\Omega_e}$ evaluates to 0 when $i \neq j$, and integrate Equation (4.15) for each basis function, ϕ_0 and ϕ_1, to generate two equations for the unknowns ($\zeta_j^{(0)}$, $\zeta_j^{(1)}$).

$$\frac{\partial}{\partial t}\Delta x_j \zeta_j^{(0)} + \frac{1}{2}[2hu_{j+1/2} - a(\zeta_{j+1}^{(0)} - 6\zeta_{j+1}^{(1)} - \zeta_j^{(0)} - 6\zeta_j^{(1)})] -$$

$$\frac{1}{2}[2hu_{j-1/2} - a(\zeta_j^{(0)} - 6\zeta_j^{(1)} - \zeta_{j-1}^{(0)} - 6\zeta_{j-1}^{(1)})] = 0 \text{ and} \quad \textbf{(4.16)}$$

$$12\Delta x_j \frac{\partial \zeta_j^{(1)}}{\partial t} - 6h(u_{j-1/2} + u_{j+1/2}) + 3[2hu_{j+1/2} - a(\zeta_{j+1}^{(0)} - 6\zeta_{j+1}^{(1)} - \zeta_j^{(0)} - 6\zeta_j^{(1)})] +$$

$$3[2hu_{j-1/2} - a(\zeta_j^{(0)} - 6\zeta_j^{(1)} - \zeta_{j-1}^{(0)} - 6\zeta_{j-1}^{(1)})] = 0 \quad \textbf{(4.17)}$$

where the boundary information has been taken from the equations in Figure 4.4, and again the interfaces are kept distinct for ease of boundary condition implementation. Explicit time stepping is used for the temporal discretization, such that the fully discretized approximations for the DG elevations with piecewise linear bases are given as

$$\frac{\Delta x_j}{\Delta t}(\zeta_j^{(0)k+1} - \zeta_j^{(0)k}) + \frac{1}{2}[2hu_{j+1/2}^k - a(\zeta_{j+1}^{(0)k} - 6\zeta_{j+1}^{(1)k} - \zeta_j^{(0)k} - 6\zeta_j^{(1)k})] -$$

$$\frac{1}{2}[2hu_{j-1/2}^k - a(\zeta_j^{(0)k} - 6\zeta_j^{(1)k} - \zeta_{j-1}^{(0)k} - 6\zeta_{j-1}^{(1)k})] = 0 \text{ and} \quad \textbf{(4.18)}$$

$$\frac{12\Delta x_j}{\Delta t}(\zeta_j^{(1)k+1} - \zeta_j^{(1)k}) - 6h(u_{j-1/2}^k + u_{j+1/2}^k) + 3[2hu_{j+1/2}^k - a(\zeta_{j+1}^{(0)k} - 6\zeta_{j+1}^{(1)k} -$$

$$\zeta_j^{(0)k} - 6\zeta_j^{(1)k})] + 3[2hu_{j-1/2}^k - a(\zeta_j^{(0)k} - 6\zeta_j^{(1)k} - \zeta_{j-1}^{(0)k} - 6\zeta_{j-1}^{(1)k})] = 0 \ . \qquad \textbf{(4.19)}$$

4.3.2 Interpolation of DG elevations to CG grid space

After the DG elevations have been calculated, they must be interpolated to the CG grid space so that the velocities can be computed using a continuous elevation field. In this study, two methods for interpolating the element average DG field to the nodal CG field are examined: 1) L_2 interpolation and 2) interface averaging. Note that when piecewise constant approximations are used for the DG elevations, the L_2 interpolation gives the same result as simple interface averaging. Refer to Figure 4.2 for grid notation.

L_2 interpolation

This type of interpolation uses weighted residuals; the general form is

$$((\zeta^{DG} - \zeta^{CG}),\omega_i)_\Omega = 0 \text{ or } (\zeta^{CG},\omega_i)_\Omega = (\zeta^{DG},\omega_i)_\Omega \qquad \textbf{(4.20)}$$

where the weighting functions, ω_i, are from the CG approximating space and the integration is over the entire domain, Ω. The left-hand side (LHS), which represents the interpolated CG values, is lumped to allow a de-coupled solution. Equation (4.20) can be rewritten as a sum of element integrals to give

$$\sum_e [(\zeta^{CG},\omega_i)_{\Omega_e} = (\zeta^{DG},\omega_i)_{\Omega_e}] \, . \qquad \textbf{(4.21)}$$

Piecewise constant DG approximation

The constant approximation for the DG elevation and the linear approximation for the CG elevation are given as

$$\zeta^{DG} = \zeta_j^{(0)}\phi_0 \text{ and } \zeta^{CG} = \zeta_{j-1/2}\omega_0 + \zeta_{j+1/2}\omega_1 \tag{4.22}$$

where the basis functions are the same as those given in §4.3.1. Substituting these approximations into Equation (4.21) gives

$$\sum_e [((\zeta_{j-1/2}\omega_0 + \zeta_{j+1/2}\omega_0),\omega_i)_{\Omega_e} = (\zeta_j^{(0)}\phi_0,\omega_i)_{\Omega_e}]. \tag{4.23}$$

Integration on a master element produces

$$\Delta x_j \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} \zeta_{j-1/2} \\ \zeta_{j+1/2} \end{bmatrix} = \frac{\Delta x_j \zeta_j^{(0)}}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{4.24}$$

where the LHS has already been lumped. Because of the discontinuities on the right-hand side, the global matrix cannot be formed in the same way as the one on the LHS. In fact, the CG global matrix will be $n \times n$ with the local element blocks overlapping; but the DG global matrix on the right will be $n \times (n-1)$ since each element is independent of its neighbors and the local element blocks will not overlap (n is the number of nodes in the domain and n-1 is the number of elements).

For a general interior node, this gives

$$\frac{\Delta x_{j-1} + \Delta x_j}{2}\zeta_{j-1/2} = \frac{\Delta x_{j-1}}{2}\zeta_{j-1}^{(0)} + \frac{\Delta x_j}{2}\zeta_j^{(0)}, \tag{4.25}$$

which for constant Δx becomes

$$\zeta_{j-1/2} = \frac{1}{2}(\zeta_{j-1}^{(0)} + \zeta_j^{(0)}). \tag{4.26}$$

This final approximation for constant Δx is the same as the arithmetic average of the interface values, as will be seen in the next main section. The constant Δx form will be used

in all subsequent sections, with the exception of the qualitative subdomain interface study in §4.8, since that application also has variable grid spacing.

Piecewise linear DG approximation

The linear approximations for CG and DG elevations as given in Equations (4.8) and (4.12) are repeated here:

$$\zeta^{CG} = \zeta_{j-1/2}\omega_0 + \zeta_{j+1/2}\omega_1 \tag{4.27}$$

$$\zeta^{DG} = \zeta_j^{(0)}\phi_0 + \zeta_j^{(1)}\phi_1 \tag{4.28}$$

where the coefficients from the CG velocity approximation have been changed to give a continuous elevation approximation. Substituting these approximations into Equation (4.21) gives

$$\sum_e [((\zeta_{j-1/2}\omega_0 + \zeta_{j+1/2}\omega_0),\omega_i)_{\Omega_e} = ((\zeta_j^{(0)}\phi_0 + \zeta_j^{(1)}\phi_1),\omega_i)_{\Omega_e}] \tag{4.29}$$

where the basis functions are as given in §4.3.1. Integration on a master element produces

$$\Delta x_j \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} \zeta_{j-1/2} \\ \zeta_{j+1/2} \end{bmatrix} = \Delta x_j \begin{bmatrix} 0.5 & -1 \\ 0.5 & 1 \end{bmatrix} \begin{bmatrix} \zeta_j^{(0)} \\ \zeta_j^{(1)} \end{bmatrix} \tag{4.30}$$

where the LHS has already been lumped. Again the global matrix on the right-hand side cannot be formed in the same way as the one on the LHS. For a piecewise linear DG approximation, the CG global matrix will be $n \times n$ with the local element blocks overlapping; but the DG global matrix on the right will be $n \times 2(n-1)$, where the column dimension is the product of the number of elements and the number of unknowns per element.

For a general interior node this gives

$$\frac{\Delta x_{j-1}+\Delta x_j}{2}\zeta_{j-1/2} = \frac{\Delta x_{j-1}}{2}\zeta_{j-1}^{(0)} + \Delta x_{j-1}\zeta_{j-1}^{(1)} + \frac{\Delta x_j}{2}\zeta_j^{(0)} - \Delta x_j\zeta_j^{(1)}, \tag{4.31}$$

which for constant Δx simplifies to

$$\zeta_{j-1/2} = \frac{1}{2}(\zeta_{j-1}^{(0)} + \zeta_j^{(0)}) + (\zeta_{j-1}^{(1)} - \zeta_j^{(1)}). \tag{4.32}$$

The interpolation formula for the first node (the ocean boundary) is taken from the first row of the matrix in (4.30) and is given as

$$\zeta_{1/2} = \zeta_1^{(0)} - 2\zeta_1^{(1)} \tag{4.33}$$

and the last node at the land boundary is taken from the second row of the matrix in (4.30) and is given as

$$\zeta_{n-1/2} = \zeta_{n-1}^{(0)} + 2\zeta_{n-1}^{(1)}. \tag{4.34}$$

Again, the constant Δx form is used in all subsequent sections, with the exception of the subdomain interface study in §4.8.

Interface averaging

This simple scheme uses the average of the DG values at the element interface to compute a CG nodal value. On the interior nodes, the CG elevation value for a piecewise constant DG approximation is calculated as

$$\zeta_{j\pm 1/2} = (\zeta_{j\pm 1} + \zeta_j)/2 \tag{4.35}$$

while for a piecewise linear DG approximation the CG nodal value is calculated as

$$\zeta_{j\pm 1/2} = [(\zeta_{j\pm 1}^{(0)} \mp 6\zeta_{j\pm 1}^{(1)}) + (\zeta_j^{(0)} \pm 6\zeta_j^{(1)})]. \tag{4.36}$$

At the ocean boundary, the CG nodal value is just the specified boundary condition, $\hat{\zeta}$. At the land boundary the nodal value for a piecewise constant DG approximation is given as

$$\zeta_{n-1/2} = \zeta_{n-1}, \tag{4.37}$$

while for a piecewise linear DG approximation the CG nodal value at the land boundary is given as

$$\zeta_{n-1/2} = \zeta_{n-1}^{(0)} + 6\zeta_{n-1}^{(1)}. \tag{4.38}$$

4.3.3 CG solution of the momentum equation

The final step in the DG/CG discretization of the primitive SWE is to discretize the momentum equation using continuous Galerkin finite elements. Begin by multiplying the momentum equation by a continuous weight function, $\omega_i(x)$, and then integrate over the domain, Ω, to get

$$\left(\frac{\partial u}{\partial t} + \tau u + g\frac{\partial \zeta}{\partial x}, \omega_i\right)_{\Omega} = 0 \tag{4.39}$$

where the explicit functional dependence of the variables, $u(x,t)$ and $\zeta(x,t)$, and the weight function has been left off for brevity. Equation (4.39) can be rewritten as a sum of element integrals to give

$$\sum_e \left[\left(\frac{\partial u}{\partial t} + \tau u + g\frac{\partial \zeta}{\partial x}, \omega_i\right)_{\Omega_e} = 0\right]. \tag{4.40}$$

Now approximate the elevation and velocity by continuous linear functions so that

$$u^h = u_{j-1/2}\omega_0 + u_{j+1/2}\omega_1 \text{ and} \tag{4.41}$$

$$\zeta^h = \zeta_{j-1/2}\omega_0 + \zeta_{j+1/2}\omega_1 \tag{4.42}$$

where the basis functions are standard Lagrange polynomials

$$\omega_0 = \frac{x - x_{j+1/2}}{x_{j-1/2} - x_{j+1/2}} \text{ and } \omega_1 = \frac{x - x_{j-1/2}}{x_{j+1/2} - x_{j-1/2}}. \tag{4.43}$$

Substituting the approximations (4.41) and (4.42) into Equation (4.40) gives

$$\sum_e \Big[\frac{\partial}{\partial t}((u_{j-1/2}\omega_0 + u_{j+1/2}\omega_1), \omega_i)_{\Omega_e} + \tau((u_{j-1/2}\omega_0 + u_{j+1/2}\omega_1), \omega_i)_{\Omega_e} +$$

$$g(\frac{\partial}{\partial x}(\zeta_{j-1/2}\omega_0 + \zeta_{j+1/2}\omega_1), \omega_i)_{\Omega_e} = 0 \Big], \tag{4.44}$$

where shorthand notation can be used to write Equation (4.44) as

$$\sum_e \Big[\frac{\partial u_j}{\partial t}(\omega_j, \omega_i)_{\Omega_e} + \tau u_j(\omega_j, \omega_i)_{\Omega_e} + g\zeta_j(\frac{\partial \omega_j}{\partial x}, \omega_i)_{\Omega_e} = 0 \Big]. \tag{4.45}$$

Integration on a master element produces

$$M_{i,j} = (\omega_j, \omega_i)_{\Omega_e} = \Delta x_j \begin{bmatrix} 1/3 & 1/6 \\ 1/6 & 1/3 \end{bmatrix} \text{ and} \tag{4.46}$$

$$B_{i,j} = (\frac{\partial \omega_j}{\partial x}, \omega_i)_{\Omega_e} = \begin{bmatrix} -0.5 & 0.5 \\ -0.5 & 0.5 \end{bmatrix} \tag{4.47}$$

such that on an element the approximation becomes

$$M_{i,j}\frac{\partial u_j}{\partial t} + \tau M_{i,j}u_j + gB_{i,j}\zeta_j = 0. \tag{4.48}$$

Following the derivation of the production code, ADCIRC, a lumped two-level time

discretization is used for the momentum equation (Crank-Nicolson). This gives

$$\frac{1}{\Delta t}M^L_{i,j}(u_j^{k+1}-u_j^k)+\frac{\tau}{2}M^L_{i,j}(u_j^{k+1}+u_j^k)+\frac{g}{2}B_{i,j}(\zeta_j^{k+1}+\zeta_j^k) \;=\; 0 \qquad \textbf{(4.49)}$$

where the lumped mass matrix is given by

$$M^L_{i,j} \;=\; \Delta x_j \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \qquad \textbf{(4.50)}$$

and the temporal terms can be gathered to give

$$\left(1+\frac{\tau\Delta t}{2}\right)M^L_{i,j}\,u_j^{k+1} \;=\; \left(1-\frac{\tau\Delta t}{2}\right)M^L_{i,j}\,u_j^k-\frac{g\Delta t}{2}B_{i,j}(\zeta_j^{k+1}+\zeta_j^k)\,. \qquad \textbf{(4.51)}$$

Standard procedures are used to build the global mass matrix on the LHS, which is diagonal due to the lumping, as well as the RHS vector. The general equation for an interior node is found to be

$$\left(1+\frac{\tau\Delta t}{2}\right)\Delta x_j\,u^{k+1}_{j-1/2}=\left(1-\frac{\tau\Delta t}{2}\right)\Delta x_j u^k_{j-1/2}+\frac{g\Delta t}{4}(\zeta^{k+1}_{j-3/2}+\zeta^k_{j-3/2}-\zeta^{k+1}_{j+1/2}-\zeta^k_{j+1/2}) \qquad \textbf{(4.52)}$$

which is valid for variable or constant Δx since the mass matrix was lumped.

4.3.4 Boundary and initial conditions

For a tidal simulation with open ocean on one end and closed land on the other end, the boundary and initial conditions are specified as

$\zeta \;=\; \hat{\zeta}(t)$ and $u_x = 0$ for $x = 0$, $t > 0$

$\zeta_x = 0$ and $u = 0$ for $x = L$, $t > 0$

$\zeta = 0$ and $u = 0$ for $x = [0, L]$, t=0

where L is the length of the domain and subscripts denote differentiation.

For the DG discretization, the tidal BC at the open ocean is enforced by setting $\zeta_L = \hat{\zeta}(t)$ at x=0. In the piecewise constant approximation, this amounts to replacing ζ_{j-1} in Equation (4.10) with the boundary condition; while in the piecewise linear approximation the term $(-\zeta_{j-1}^{(0)} - 6\zeta_{j-1}^{(1)})$ in Equations (4.16) and (4.17) is replaced with $-\hat{\zeta}(t)$. The land BC at x=L is enforced in the piecewise constant approximation by setting $(\zeta_{j+1} - \zeta_j) = 0$ in Equation (4.10) and it is enforced in the linear approximation by setting $(\zeta_{j+1}^{(0)} - 6\zeta_{j+1}^{(1)} - \zeta_j^{(0)} - 6\zeta_j^{(1)}) = 0$ in Equations (4.16) and (4.17).

For the CG discretization, the BC at the open ocean does not require any modification to the global momentum matrix. The land BC is enforced by replacing the last row in the global matrix with the equation $u_n = 0$, where n is the index of the last node in the domain.

4.4 Method 2: DG <–> ADCIRC subdomain coupling

In this method, the primitive system of SWE is discretized using DG finite elements in one subdomain and the ADCIRC formulation is discretized using CG finite elements in another subdomain. Information is passed between these subdomains by using overlapping elements. In the DG subdomain, the discrete equations for piecewise constants are as given by Equations (B.42) and (B.43) in Appendix B, while the equations for piecewise linear Legendre interpolants are as given by Equations (B.53) through (B.56). In the ADCIRC subdomain, the discrete equations are as given by Equations (A.24) and (A.25) in Appendix A. These equations are not repeated here, but instead a brief description of the models is given. The schematic from Figure 4.1 is repeated here as Figure 4.5 with more detail.

The DG model uses either piecewise discontinuous constants or linear Legendre approximating functions for both the elevation and velocity in a Galerkin FE discretization. A minmod slope limiter can be used for added stability with the piecewise linear approximations. The temporal discretization is a simple explicit scheme so that only information from the previous time step (time level k) is required to compute the new

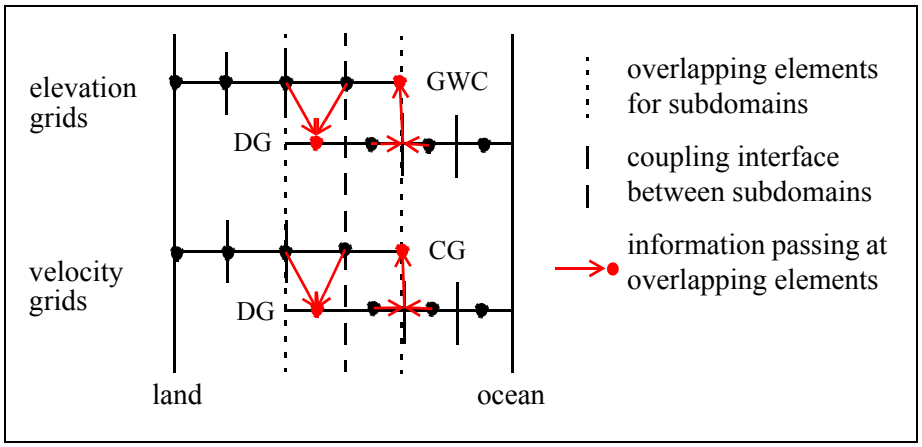


Figure 4.5 Subdomain grid notation for DG <–> ADCIRC multi-algorithmic coupling scheme.

elevation and velocity state values. The ADCIRC model uses continuous linear Lagrange approximating functions for both the elevation and the velocity in a Galerkin FE discretization. The GWC equation uses a semi-implicit three-level time stepping scheme, where information from the past two levels is required to compute the new elevation state values, which are coupled in a tridiagonal matrix in 1D (time levels k–1, k, k+1). The momentum equation uses a lumped two-level temporal scheme (time level k) so that only information from the previous time step is required to compute the new velocity state values.

With this second subdomain coupling method three main questions arise: 1) at which time levels *should* the subdomains be coupled?; 2) how *can* the subdomains be coupled at these time levels?; and 3) how should information be passed since the grid spaces are different – CG versus DG?

To answer the first question, note that the DG model presented in Appendix B is explicit. Therefore, as far as the DG subdomain is concerned, it only needs information to be passed from the ADCIRC subdomain at the k time level in preparation for the next time step. However, the ADCIRC model uses a semi-implicit, three-level time scheme for the

elevations and needs information from the past two time steps in order to calculate the new state values. Furthermore, with the semi-implicit scheme, the interior node at the subdomain interface will require information from the overlapping element at the present time level. Thus, the elevation solution in the ADCIRC subdomain needs information for the overlapping element at all three time levels (k–1, k, k+1); while the velocity solution only requires velocity information at the k time level, but requires elevation information at two time levels (k and k+1).

This brings up the second question, if explicit passing is to be used at the subdomain interface, how should the semi-implicit solution for the ADCIRC elevations be addressed? As a first approximation, a combined explicit / implicit passing scheme is utilized. Since the DG subdomain is completely explicit, the solution can be computed without any information from the current time level (k+1) at the subdomain interface. Then the overlapping nodal elevation information can be passed to ADCIRC at all three time levels (k–1, k, k+1); and the ADCIRC solution can proceed with all of the information that it requires. Finally, the information for the overlapping element at the subdomain interface is passed from the ADCIRC subdomain to the DG subdomain in preparation for the next time step. With this explicit / implicit subdomain passing scheme, the ADCIRC subdomain is coupled in all three time levels, the DG subdomain is coupled in the past time level, and both subdomains have all of the information that they need for a solution. Additional care must be taken to ensure that the elevation information in the DG subdomain is not over-written before the old levels have been passed into the ADCIRC subdomain.

Finally, to address the third question, note that at the overlapping elements the same problem that was encountered in the DG/CG equation coupling – namely, grid communication between CG and DG grid spaces – becomes an issue. For the DG to CG passing, both L_2 interpolation and interface averaging were examined, as in coupling method 1. This is shown schematically in Figure 4.5 by the red arrows that transfer the element centered DG information to the element interface and then pass this interface

information to the CG node.

For the CG to DG passing, element averages are used. This is shown schematically in Figure 4.5 by the red arrows that use the nodal CG information to pass element centered information to the DG element. Since both the CG and DG approximating spaces are linear, an element slope and average can easily be computed from the CG nodal values. The element average for the overlapping element is simply the average of the nodal values located at the element edges, while the slope is computed by dividing their difference by Δx. However, recall that the slope in the DG framework contains a multiplying factor, such that the slope for the b variable is given as slope = $12b_j^{(1)}/\Delta x$. The resulting equations for the element centered DOFs are

$$b_j^{(0)} = (b_{j-1/2} + b_{j+1/2})/2 \tag{4.53}$$

$$b_j^{(1)} = [(b_{j+1/2} - b_{j-1/2})/\Delta x](\Delta x/12) = (b_{j+1/2} - b_{j-1/2})/12 \tag{4.54}$$

where b can be either the elevation or velocity state variable and the $j \pm 1/2$ points are the element edge locations, as in coupling method 1.

The basic solution procedure for this coupled model is to initialize the entire domain and then compute the new elevations and velocities in the DG subdomain. Then the interface information is passed from the DG subdomain to the ADCIRC subdomain using overlapping elements (time levels k–1, k, and k+1 for elevations and time level k for velocity) and the new elevation and velocities are computed in the ADCIRC subdomain. Finally, the interface information is passed from the ADCIRC subdomain to the DG subdomain at the overlapping element in preparation for the next time step (time level k for both elevation and velocity). This subdomain calculation and passing process continues through all of the time steps.

4.4.1 Boundary and initial conditions

For a tidal simulation with open ocean on one end and closed land on the other end, the boundary and initial conditions are specified as

$\zeta = \hat{\zeta}(t)$ and $u_x = 0$ for $x = 0$, $t > 0$

$\zeta_x = 0$ and $u = 0$ for $x = L$, $t > 0$

$\zeta = 0$ and $u = 0$ for $x = [0, L]$, t=0

where L is the length of the domain and subscripts indicate differentiation. The initial conditions are often called a "cold start" since the elevation and velocity fields throughout the entire domain are set to zero at time, t=0.

The tidal BC at the open ocean is enforced in the ADCIRC model subdomain, since it is desirable to use the model that is more computationally efficient in the more stable, deep flow region. Likewise, the land BC is enforced in the DG model subdomain since discontinuous models are more adept at handling the shocks and highly advective flows of near-shore regions.

Thus, within the ADCIRC subdomain, the ocean BC is enforced at x=0 by replacing the first row of the GWC global matrix with $\zeta_1 = \hat{\zeta}(t)$. No modifications are required in the momentum discretization since there are no boundary flux terms in the derivation. Within the DG subdomain, the land BC is enforced at x=L by setting $u_R = 0$ in the momentum discretization and $\zeta_R = \zeta_L$ in the continuity discretization.

4.5 Method 3: DG/CG <–> ADCIRC subdomain coupling

In this method, the primitive system of SWE is discretized using the DG/CG equation coupling method in one subdomain and the ADCIRC formulation is discretized using CG finite elements in another subdomain. Information is passed between these subdomains by using overlapping elements. For piecewise linear DG approximations,

Equations (4.16), (4.17) and (4.51) from §4.3 are used in the DG/CG subdomain, and Equations (A.24) and (A.25) from Appendix A are used in the ADCIRC subdomain. (For piecewise constant DG approximations, Equation (4.11) is used instead of (4.16) and (4.17).) These equations are not repeated here; details can be found in the referenced sections. The schematic from Figure 4.1 is repeated here as Figure 4.6 with more detail.

The same three questions arise in this coupling method as were discussed for the second coupling method: 1) at which time levels *should* the subdomains be coupled?; 2) how *can* the subdomains be coupled at these time levels?; and 3) how should information be passed since the grid spaces are different – CG versus DG?

To address the first question, recall that the ADCIRC model is semi-implicit and requires information at the new time step (k+1 time level) in order to solve for the new elevations and velocities. Furthermore, due to the Crank-Nicolson time stepping in the CG momentum approximation, the DG/CG model also requires elevation data at the new time step (k+1 time level) in order to solve for the new velocities (notice in Equation (4.51) from §4.3 that ζ^{k+1} is required on the right hand side). Therefore, both subdomains require

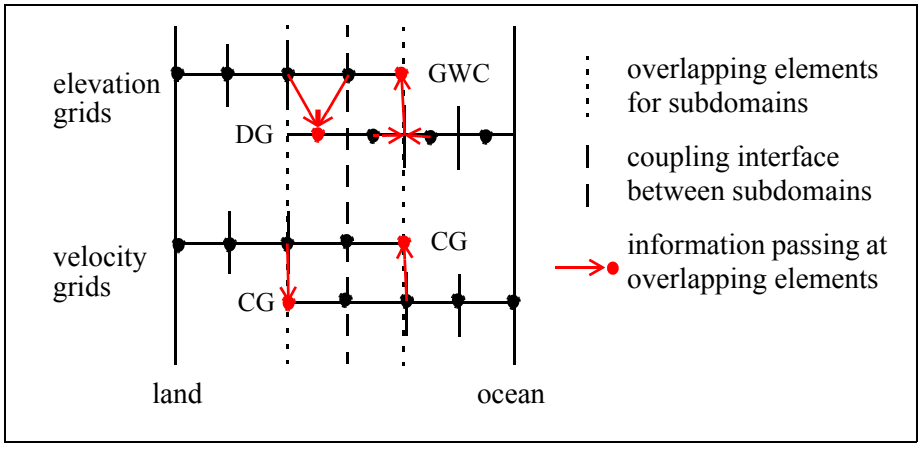


Figure 4.6 Subdomain grid notation for DG/CG <–> ADCIRC multi-algorithmic coupling scheme.

information from the current time step at the overlapping element in order to obtain a solution. This is in contrast to the DG↔ADCIRC model, where the DG subdomain is completely explicit.

This brings us to the second question; how *can* we couple the models at the required time levels? Technically, some sort of predictor-corrector scheme or explicit overlapping with iterations is required to satisfy both models. However, for this study a simple explicit/implicit scheme similar to that used for the DG↔ADCIRC model is employed instead. The reader is referred to §4.4 for more details about this scheme. Briefly, the DG/CG subdomain is solved using explicit information at the subdomain interface, then information at all three time levels is passed at the overlap to the ADCIRC subdomain, which then computes the new elevation and velocity solution with the required information and passes back to the DG/CG model at the overlap in preparation for the next time step. Essentially, the data requirements for the ADCIRC model are satisfied, while those for the velocity solution in the DG/CG model are not.

Finally, to address the third question, note first of all that the velocity approximations are continuous in both of the subdomains. Within the DG/CG subdomain itself, the DG elevations are interpolated onto the CG grid using either L_2 interpolation or interface averaging, as was discussed in detail in §4.3.2. The elevation nodal value that is calculated in this CG interpolation step can also be used to pass the elevation data from the DG grid to the CG grid at the overlapping element. Passing of elevation from the ADCIRC subdomain to the DG subdomain at the overlapping element is done as described in §4.4; briefly, the elevation element average and slope are computed from the CG nodal values and passed to the DG grid.

For the velocity state values, both subdomains employ continuous approximations; so it is possible to simply pass the nodal values at the overlapping elements directly from one model to the other without any additional calculations. This is shown by the red arrows

in Figure 4.6, which pass directly along the overlapping element interface for the velocity grids. Notice that although the velocity approximations at the physical subdomain interface are continuous, the velocity is allowed to be discontinuous at this point. This is because each subdomain calculates a new value for this node; these values will not necessarily be equal. Plots of the interface velocity for an application are presented in §4.8, and show that these differences are most often small for the cases considered herein.

4.5.1 Boundary and initial conditions

For a tidal simulation with open ocean on one end and closed land on the other end, the boundary and initial conditions are specified as

$\zeta = \hat{\zeta}(t)$ and $u_x = 0$ for $x = 0$, $t > 0$

$\zeta_x = 0$ and $u = 0$ for $x = L$, $t > 0$

$\zeta = 0$ and $u = 0$ for $x = [0, L]$, t=0

where L is the length of the domain and subscripts denote differentiation. The initial conditions are often called a "cold start" since the elevation and velocity fields throughout the entire domain are set to zero at time, t=0.

The tidal BC at the open ocean is enforced in the ADCIRC model subdomain, since it is desirable to use the model that is more computationally efficient in the more stable, deep flow region. Likewise, the land BC is enforced in the DG/CG model subdomain since discontinuous models are more adept at handling the shocks and highly advective flows in near-shore regions.

Thus, within the ADCIRC subdomain, the ocean BC is enforced at x=0 by replacing the first row of the GWC global matrix with $\zeta_1 = \hat{\zeta}(t)$. No modifications are required in the momentum discretization, since there are no boundary flux terms in the derivation. Within the DG/CG subdomain, the land BC is enforced at x=L by replacing the last row of the global matrix for momentum with $u_n = 0$ and setting $\zeta_R = \zeta_L$ in the continuity

discretization.

4.6 Propagation analysis of coupled schemes

Using the numerical propagation analysis tools developed in Chapter 2, each of the coupling methods presented above are analyzed to examine wave propagation through the subdomain interface. A constant bathymetric depth must be used for the numerical tools, so this part of the study will only determine whether the coupling methods are propagating the waves correctly. Section 4.8 will examine the location of the coupling interface relative to the bathymetric depth by qualitatively comparing the simulation results as the interface is moved.

Since the wave begins in the ADCIRC model domain, the boundary forcing initialization method is used for the numerical propagation analysis (see discussion in §2.4.3). Also, as recommended in Chapter 2, the least squares harmonic wave tracking scheme is used. The parameters for this study are the same as were used for the propagation studies in Chapters 2 and 3 and are repeated here in Table 4.1 for convenience. For coupling methods 2 and 3, the subdomain interface is located at exactly one wavelength into the domain – $k\Delta x$. (Recall that there is no subdomain interface for coupling method 1, since it is an equation coupled scheme.)

Table 4.1 Parameters used in coupled algorithm study.

$g = 9.81$ m/s^2	$h = 10$ m	$\Delta t = 1.0$ s
$\tau = 0.0001$ s^{-1}	$G = 0.001$s^{-1}	$\Delta x = 1000$ m

First, individual plots of propagation behavior are presented for each of the three coupling methods to compare the results for piecewise constant and piecewise linear interpolants in the DG algorithm. Then a full comparison of propagation behavior for the best version of each coupling method, the ADCIRC model, and domain-wide low- and

higher-order DG algorithms are presented.

Figure 4.7 presents the propagation behavior of coupling method 1 (DG discretization of the continuity equation and CG discretization of the momentum equation) using piecewise constants and piecewise linears in the DG approximation. These results were generated using L_2 interpolation of the DG elevations to the CG grid, since interface averaging was found to be unstable.

As expected, given the results in Chapter 3, the piecewise linear approximations

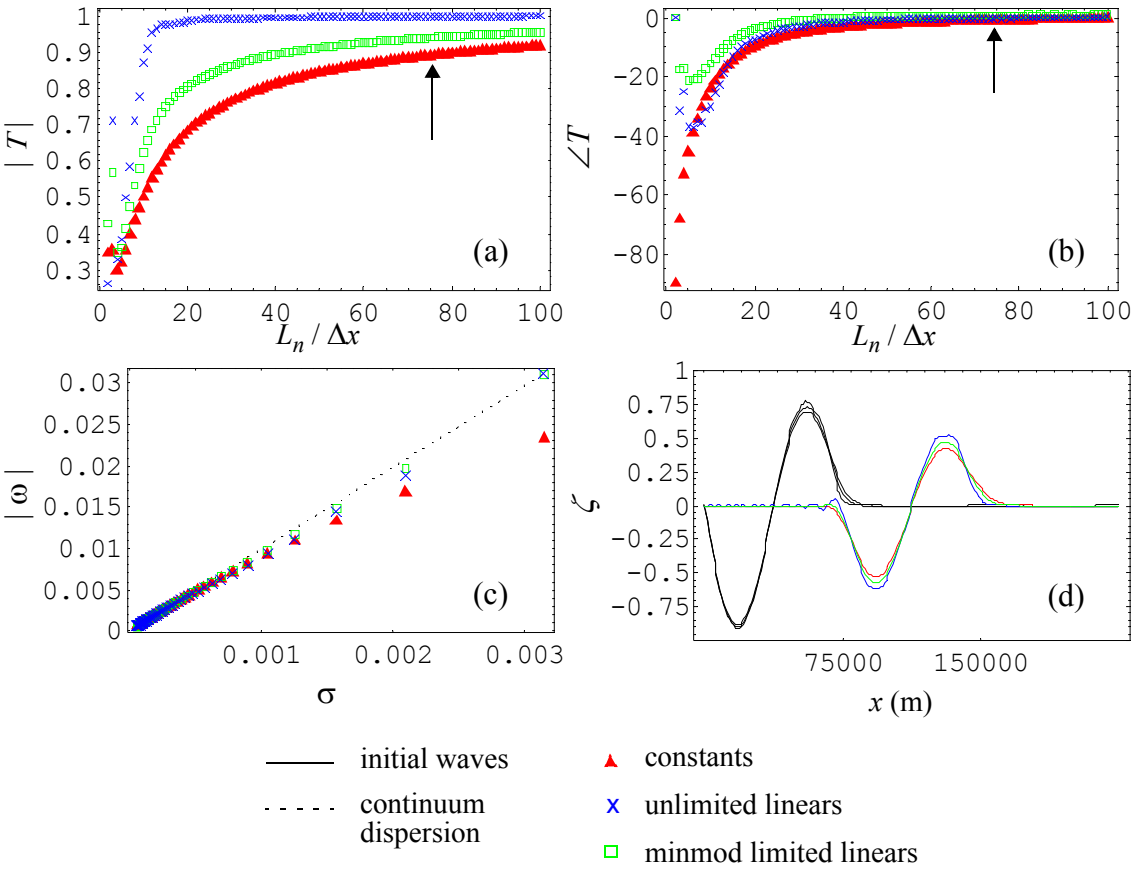


Figure 4.7 Comparison of propagation characteristics for DG/CG model with various DG spatial approximations: (a) damping ratio, (b) phase error, (c) dispersion relation, (d) elevation profiles for $75\Delta x$ wave. (Arrows show location of data points generated from the profiles in panel d.)

result in less damping of the physical wavelengths. Notice that the phase error for all three approximations approaches an asymptotic value of zero. Meanwhile, the damping ratio for the low-order (piecewise constants) scheme only approaches a value of 0.92 while the minmod limited linear approximation approaches a value of 0.96 and the unlimited linear approximation reaches the desired value of unity. Thus the minmod limited approximation is less dissipative than the low-order and more dissipative than the unlimited linear approximation. The dispersion curves for the higher-order approximations match the continuum relationship well, while the curve for the piecewise constant approximation lies below the continuum relationship but remains monotonic.

Recall that information from the elevation profiles shown in panel (d), as well as the velocity results that are not shown, are used to determine the Fourier and dispersion behavior at the $75\Delta x$ wavelength. Each discrete data point in panels (a) through (c) were derived in a similar manner from the profiles for the corresponding wavelength. (The elevations shown are from the numerical propagation analysis and are not traditional simulation results. In each of the propagation figures in this chapter, arrows will indicate which data point is being shown by the profiles.) For the elevations in Figure 4.7d, notice that while the low-order model is overly dissipative in nature, it does produce a smooth solution, while the unlimited higher-order scheme allows some trailing wiggles behind the wave. However, the minmod limited scheme removes these wiggles, which also disappear in the unlimited version when higher wavelength to grid spacing ratios ($L/\Delta x$) are used.

Figure 4.8 compares 1D elevation results from a simulation for all three approximations to a fine grid solution from a domain-wide ADCIRC simulation (Δx = 39.0625m) and verifies the phase characteristics shown in Figure 4.7. (The velocity results are similar.) The simulation parameters are the same as in Table 4.1, and the period of the open-ocean boundary condition was chosen such that the simulated wavelength satisfied $L_n/\Delta x$ equal to 30. The amplitude of the forcing function was set equal to 1.0 m. The period of the boundary condition is calculated as

$$\mathrm{T} = L_n/c = 30\Delta x \Big/ \sqrt{gh - \left(0.5\tau \frac{30\Delta x}{2\pi}\right)^2} \quad , \tag{4.55}$$

which gives T = 3029.79 seconds for the parameters given in Table 4.1. Additionally, the total domain length was set equal to 100,000m to avoid the waves reflecting at the land boundary during the simulation. Notice that the low-order scheme, using piecewise constants, significantly damps the solution as the waves progress further into the domain.

Figure 4.8b shows an expanded view of the first peak after three full periods of simulation time (10,000 seconds). The first peak was chosen since it has traveled approximately one wavelength through the domain, which should result in the best comparison with the predicted propagation results, as they were calculated on a per-wavelength basis; and the long simulation time was to allow the domain to spin up from a cold start before analyzing the results.

Table 4.2 compares the predicted propagation characteristics, taken from the phase plots in Figure 4.7 at $L_n/\Delta x = 30$, and the measured propagation characteristics, taken from the observed simulation behavior in Figure 4.8b. Notice that, despite the relative coarseness of the coupled model compared to the fine grid solution, the actual damping errors seen in the simulation are predicted quite closely by the numerical propagation tools. However, the coarseness of the simulation results (Δx = 1000m) does affect the phase error, particularly

Table 4.2 Comparison of predicted and measured propagation characteristics for DG/CG equation coupled model with various DG interpolants.

Relative to ADCIRC	Phase error in degrees (+ lead, – lag)		Damping ratio	
	Measured	Predicted	Measured	Predicted
constants	+1.125	–5.442	0.722	0.767
unlimited linears	–10.875	–3.008	1.020	0.997
minmod linears	+1.125	–1.031	0.929	0.866

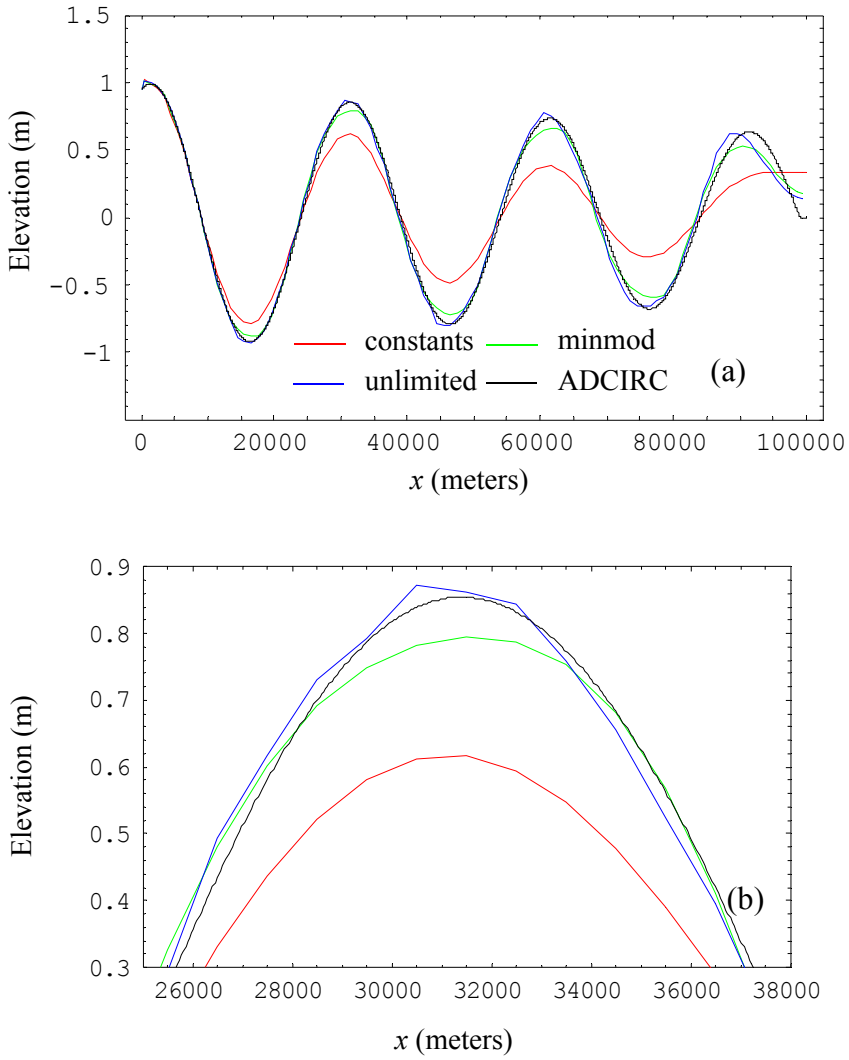


Figure 4.8 Comparison of DG/CG equation coupled model results with a fine grid ADCIRC simulation after three full periods of a $30\Delta x$ wave: (a) full domain, (b) zoomed view of first peak.

for the unlimited linear version.

Figure 4.9 presents the propagation behavior of the DG $\leftrightarrow$ ADCIRC model with three different spatial approximations in the DG domain. In the simulation output, Figure 4.9d, the vertical dashed line indicates the location of the subdomain interface.

For the damping ratios, notice that the minmod limited and unlimited results are less dissipative than the low-order and are nearly the same for wavelengths greater than $50\Delta x$.

Meanwhile, the phase errors for both of the higher-order versions are nearly equivalent and all three versions exhibit phase leads in the low wavelengths, which are damped out. However, the low-order version over-damps the longer wavelengths, even though the phase errors are similar to the higher-order versions and are near zero. The dispersion curves for all three are monotonic, although they are slightly higher than the continuum relationship, which is to be expected give the phase lead in the phase errors. There is a slight tail in the ADCIRC subdomain even after the wave has passed through the interface, particularly for

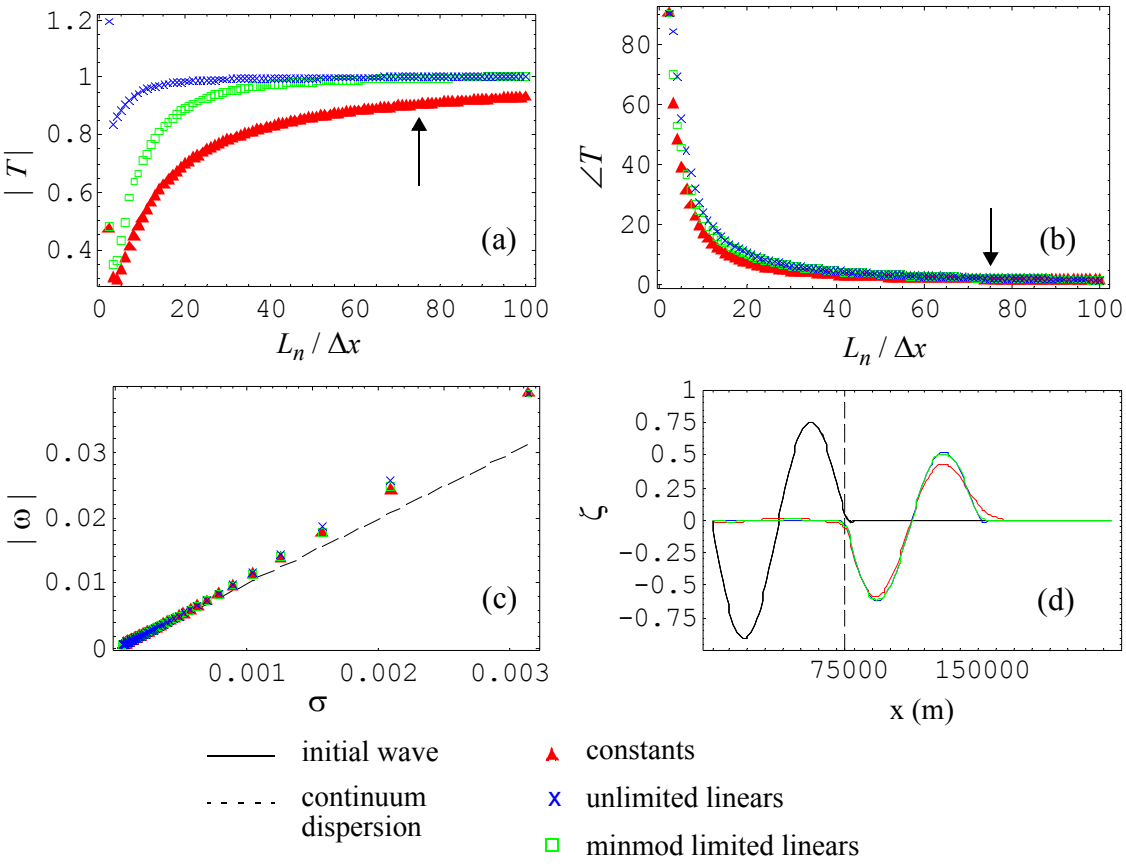


Figure 4.9 Comparison of propagation characteristics for DG <–> ADCIRC model with various DG spatial approximations: (a) damping ratio, (b) phase errors, (c) dispersion relations, and (d) elevation profiles for 75Δx wave. (Arrows show location of data points generated from the profiles in panel d.)

the low-order version. This artifact disappears as the wavelength to grid spacing ratio ($L/\Delta x$) increases and may be due to the sudden removal of the boundary forcing, since it does not exist when the waves are continually introduced at the boundary.

Figure 4.10 compares 1D elevation results from simulations using each of the three discontinuous spatial approximations to a fine grid solution from the commonly-used ADCIRC model (Δx = 39.0625m). The simulation parameters are the same as were used in the previous verification for the DG/CG model. The vertical dashed line indicates the location of the subdomain interface; notice that the waves pass through this interface without significant distortion.

Table 4.3 compares the predicted propagation characteristics, taken from the phase plots in Figure 4.9 at $L_n/\Delta x = 30$, and the measured propagation characteristics, taken from the observed simulation behavior in Figure 4.10b. Since the first peak lies at the subdomain interface, the second peak is used for the comparison, and the damping is adjusted to include the two wavelength travel distance. This is a more valid comparison to the numerical propagation results, since in that analysis the wave was allowed to travel exactly one wavelength in each subdomain. Notice that the damping effects of the low-order approximation becomes more evident as the waves travel further into the domain, such that the measured damping is more significant than was predicted. Also note that the phase errors are affected by the coarseness of the coupled model's grid resolution.

Table 4.3 Comparison of predicted and measured propagation characteristics for the DG <–> ADCIRC model with various DG interpolants.

Relative to ADCIRC	Phase error in degrees (+ lead, – lag)		Damping ratio	
	Measured	Predicted	Measured	Predicted
constants	+ 1.125	+ 4.442	0.514	0.656
unlimited linears	+ 1.125	+ 6.650	1.014	1.007
minmod linears	+ 1.125	+ 6.453	0.862	0.875

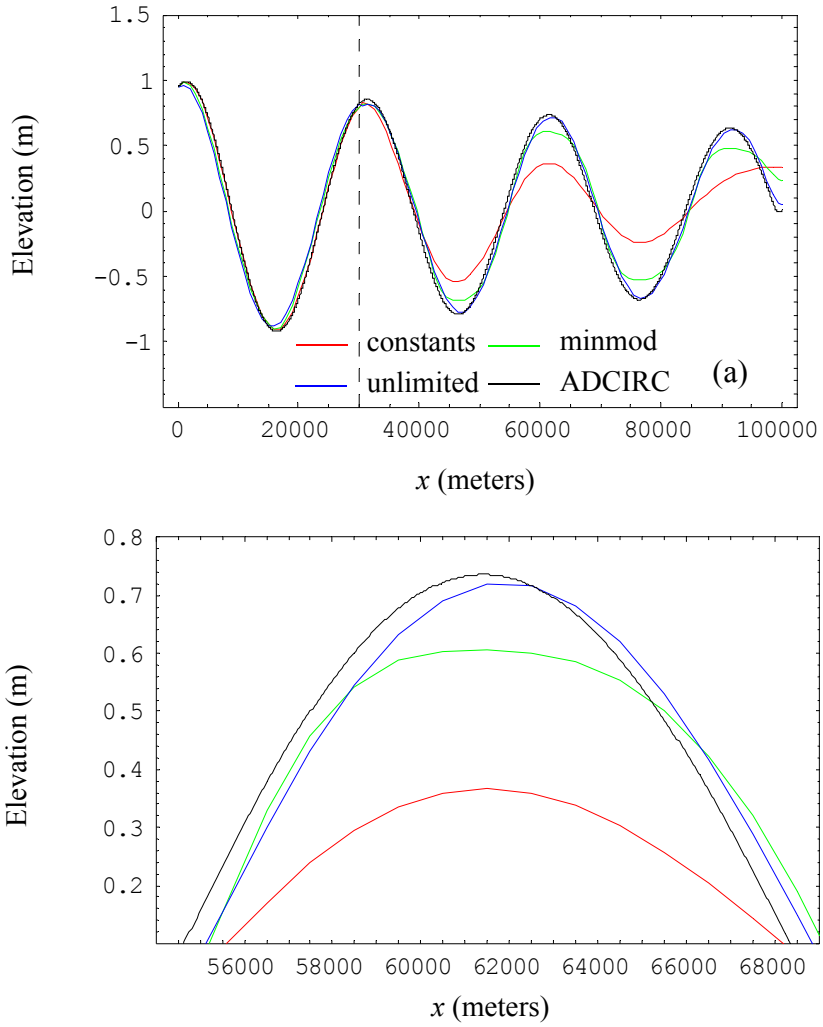


Figure 4.10 Comparison of DG <–> ADCIRC model results with a fine grid ADCIRC simulation after three full periods of a $30\Delta x$ wave: (a) full domain, (b) zoomed view of second peak.

Figure 4.11 presents the propagation behavior of the DG/CG $\leftrightarrow$ ADCIRC model using piecewise constants and piecewise linears in the DG approximation. In the simulation output, Figure 4.11d, the vertical dashed line indicates the subdomain interface. For this model, the phase errors and dispersion curves are comparable for both the low- and higher-order DG approximations. The phase error at the $100\Delta x$ wavelength ranges from 0.9 degrees for the low-order approximation to 0.7 degrees for the higher-order

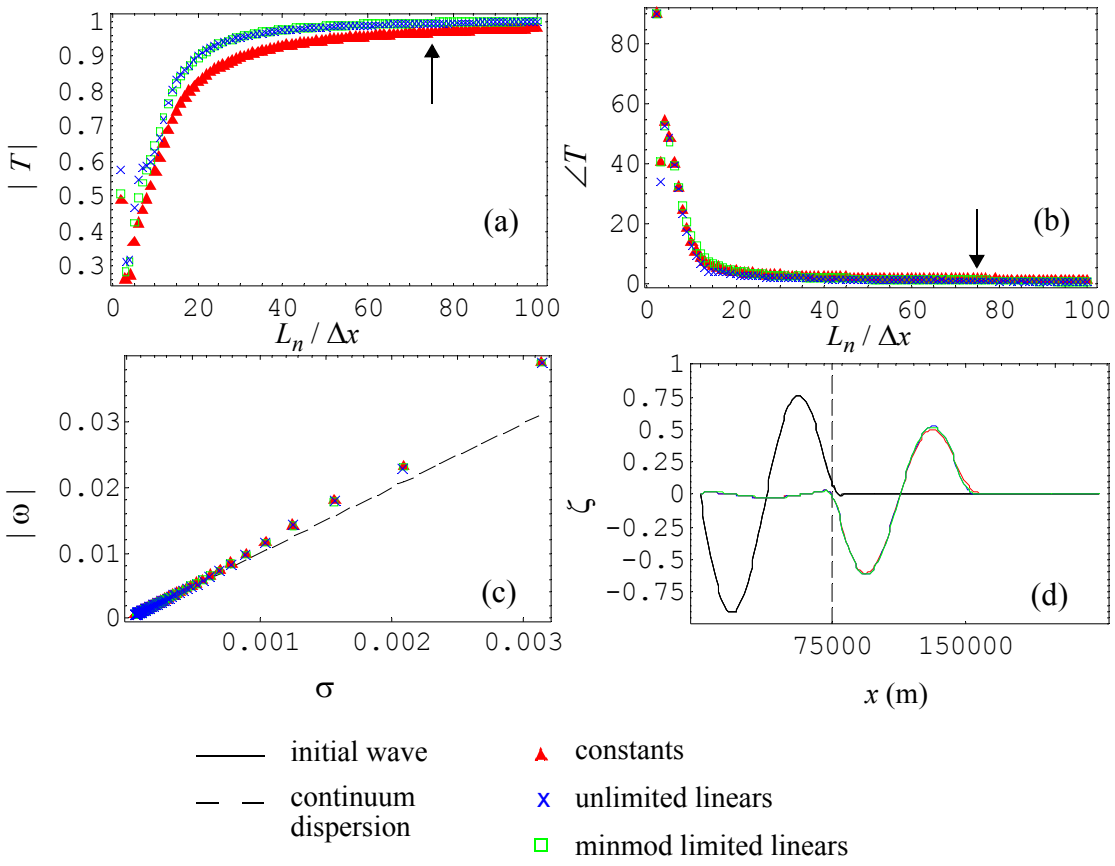


Figure 4.11 Comparison of propagation characteristics for the DG/CG <–> ADCIRC model with various DG spatial approximations: (a) damping ratio, (b) phase error, (c) dispersion relation and (d) elevation profiles for 75Δ*x* wave. (Arrows show location of data points generated from the profiles in panel d.)

approximations. However, the low-order approximation is slightly more dissipative and damps the physical wavelengths, even though they have minimal phase error.

Figure 4.12 compares 1D elevation results (the velocity results are similar) from simulations using each of the three discontinuous spatial approximations to a fine grid solution from the commonly-used ADCIRC model (Δx = 39.0625m). The simulation parameters are the same as were used to verify the previous models, and the vertical dashed line indicates the location of the subdomain interface. Notice that the waves do incur some

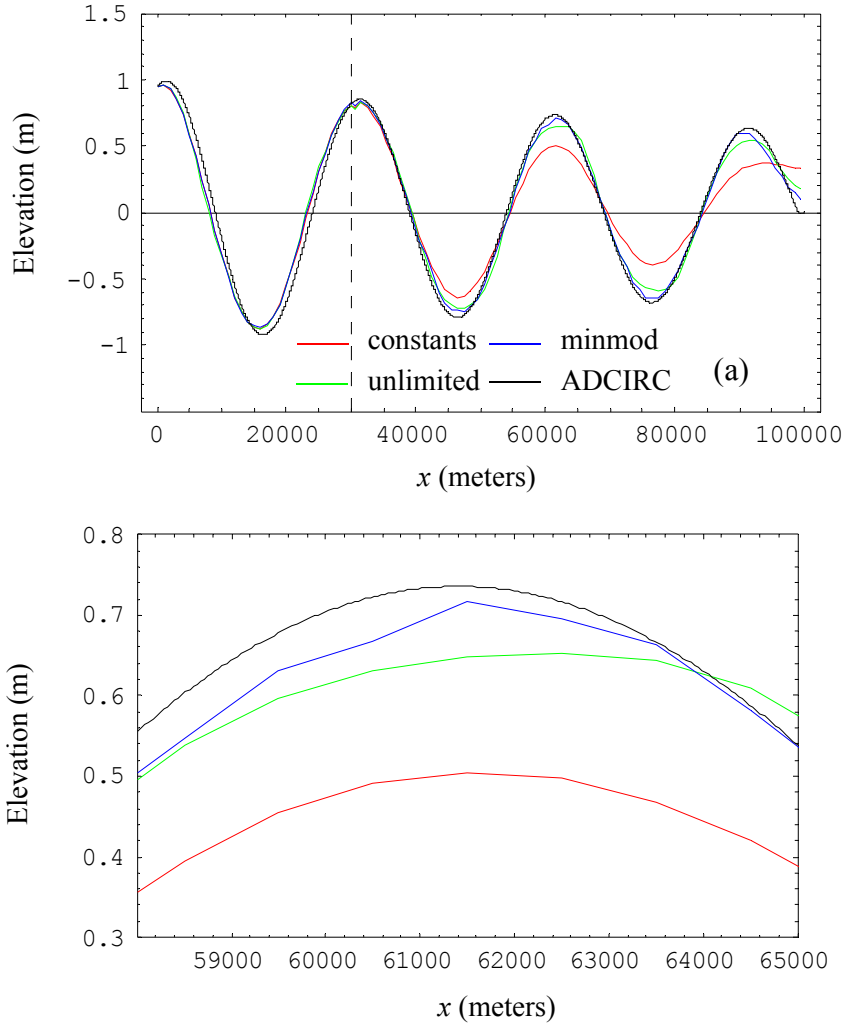


Figure 4.12 Comparison of DG/CG <–> ADCIRC subdomain coupled model results with a fine grid ADCIRC simulation after three full periods of a 30Δx wave: (a) full domain, (b) zoomed view of second peak.

distortion as they pass through the interface, which is smoothed out as they travel through the domain. Note also that the low-order version significantly damps the solution as the wave progresses through the domain.

Table 4.4 compares the predicted propagation characteristics, taken from the phase plots in Figure 4.11 at $L_n/\Delta x = 30$, and the measured propagation characteristics, taken from

the observed simulation behavior in Figure 4.12b. Since the first peak lies at the subdomain interface, the second peak is used for this comparison, and the damping is adjusted for the extra wavelength. This is a more valid comparison to the numerical propagation results, since in that analysis the wave was allowed to travel exactly one wavelength in each subdomain. Notice that, despite the relative coarseness of the coupled model compared to the fine grid solution, the actual errors seen in the simulation are predicted quite closely by the numerical propagation tools, with the exception of the phase error for the minmod limited version.

Table 4.4 Comparison of predicted and measured propagation characteristics for the DG/CG <–> ADCIRC model with various DG interpolants.

Relative to ADCIRC	Phase error in degrees (+ lead, – lag)		Damping ratio	
	Measured	Predicted	Measured	Predicted
constants	+ 1.125	+ 2.453	0.706	0.859
unlimited linears	+ 1.125	+ 2.010	0.991	0.999
minmod linears	+ 13.125	+ 2.512	0.918	0.948

Figure 4.13 compares all three coupling models with a domain-wide DG algorithm and a domain-wide ADCIRC model. Based upon the overly dissipative behavior of the low-order schemes and the similar behavior of the unlimited and minmod limited higher-order schemes, unlimited piecewise linear approximations were used for all of the coupling models as well as the domain-wide DG simulation for this comparison. Notice that the simulation results are very close for the five models.

For the Fourier propagation results, the damping behavior of coupled models 1 and 3 more closely resemble the domain-wide DG model, although they are not an exact match, while the behavior for coupled model 2 more closely resembles the ADCIRC model. Meanwhile, the phase error for coupled model 3 nearly overlies that of the ADCIRC model, while coupled model 2 has similar behavior to the domain-wide DG model but is not an

exact match to either. The phase error for coupled model 1 displays a phase lag in the low wavelengths, while all of the other four models exhibit a phase lead.

The dispersion curve for coupled model 1 most closely matches the continuum dispersion relationship, while coupled model 3 more closely matches the domain-wide DG model. The dispersion curve for coupled model 2 lies above both the ADCIRC and domain-wide DG models for all but the last data point, where it matches the DG model. In general, all five models are monotonic and exhibit acceptable dispersion behavior. From the

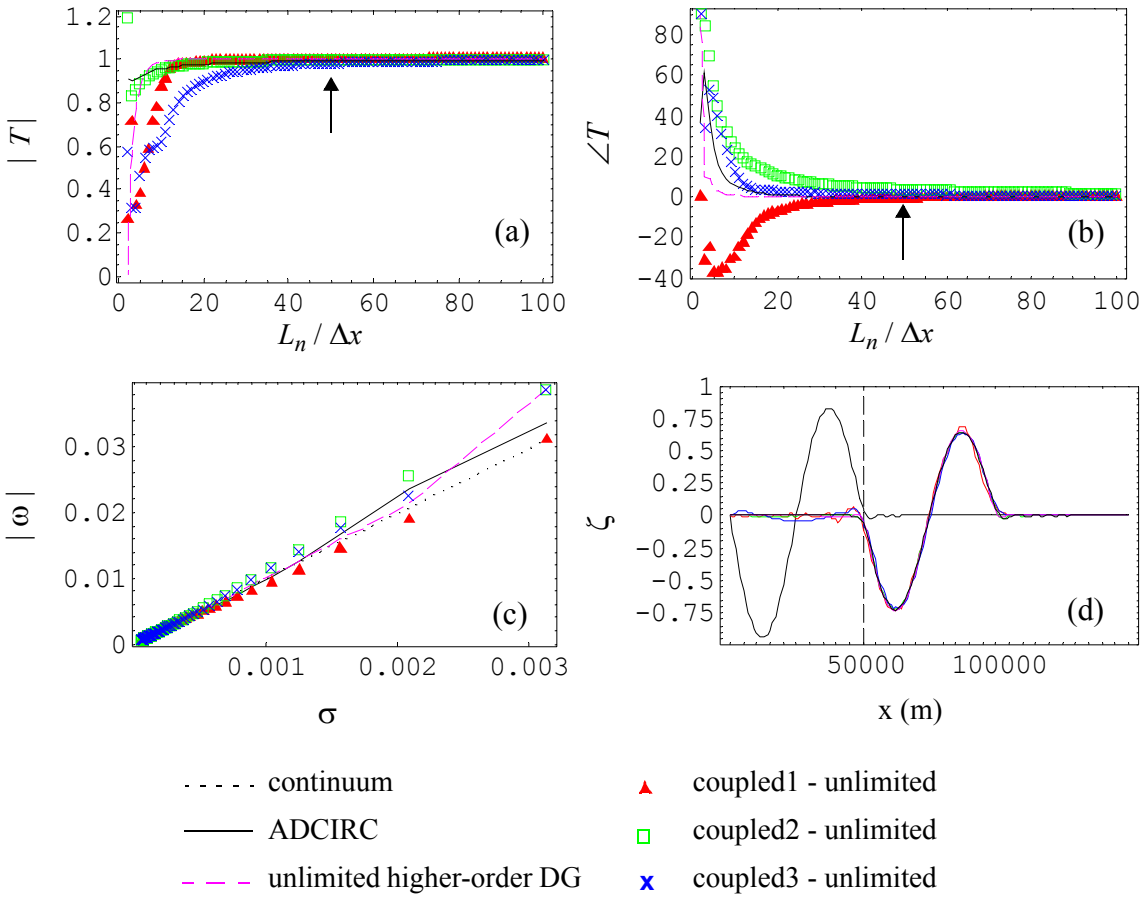


Figure 4.13 Comparison of propagation characteristics for all three coupling methods with a domain-wide DG model and the ADCIRC model: (a) damping ratio, (b) phase error, (c) dispersion relation, (d) elevation profiles for 50Δ*x* wave. (Arrows show location of data points generated from the profiles in panel d.)

propagation results, any of the coupling schemes appear to be promising for the linear model problem. However, coupled model 2 may be the most promising, as it has the smoothest transition at the interface without introducing oscillations in either subdomain. Further comparison of these five models will be made in the next section, where convergence rates are presented for a tidal application.

4.7 Numerical error analysis

4.7.1 Spatial accuracy

For the spatial convergence study, the test case was a 1D channel with M_2 tidal forcing (period = 44,712 sec.) of amplitude equal to 1m on the ocean boundary and a land boundary on the other end. The channel dimensions are 50km by 10m. The initial condition was a cold start, where the elevation and velocity fields are zero at time t=0; and the simulations were run out for slightly more than two full periods of the tidal forcing for a total of 90,000 seconds. Since the initial condition for the domain is a cold start for both the elevation and velocity fields, only the last time output is used to compute the errors. This allows the entire domain to spin-up before examining the output.

A fine grid solution (1281 nodes, $\lambda/\Delta x = 442.85$) from the ADCIRC model was used for the "true" solution. In order to isolate the spatial errors, a time step of 0.01 seconds was used for all of the simulations, including the fine grid. Two error norms were computed for the final output at t_f = 90,000 seconds: 1) an L_2 error norm in space, computed as

$$L_2 \text{ error } = \left(\left(\sum_{j=1}^{nn} \left(B(x_j, t_f) - b_j(t_f) \right)^2 \right) / nn \right)^{1/2} ; \qquad \textbf{(4.56)}$$

and 2) an L_∞ error norm in space, computed as

$$L_\infty \text{ error } = max[(B(x_j, t_f) - b_j(t_f))^2]^{1/2}, \qquad \textbf{(4.57)}$$

where nn is the number of nodes in the approximation, b (B) can be either elevation or velocity, the $b_j(t_f)$ are the simulation results at node j and time t_f=90,000 seconds, and the true solutions $B(x, t)$ are interpolated functions from the fine-grid ADCIRC solution. When these error norms are plotted against the spatial resolution in log-log space, the slope represents the accuracy (asymptotic convergence rate) of the algorithm.

A graphical presentation of the results provides for easier comparison between algorithms, therefore only the log-log plots will be presented in this section. Each of the figures will present both of the error norms for the elevation and velocity results for all three DG approximations – piecewise constant, piecewise linear with minmod limiting and piecewise linear with no limiting. The tabular data used to generate these plots is available to the interested reader in Appendix E. Also, the spatial accuracy of each algorithm, as computed from the slope of the log-log plots, is summarized in Table 4.5 on page 166 at the end of this section. Plots for each individual algorithm will be shown first, and then all of the coupled algorithms will be compared with the domain-wide DG and ADCIRC model results.

Spatial convergence results for the equation coupled DG/CG model, are presented in Figure 4.14. As was seen in the previous chapter, the higher-order DG approximations are more sensitive to stability constraints; and for the smallest grid spacing (Δx = 39.0625m) the error plots for the unlimited version begin to diverge as the model tends toward instability (use of a smaller time step brings these errors back into line). However, the minmod limited higher-order version keeps the algorithm stable at this grid spacing. Additionally, note that the error norms for the higher-order schemes are lower in magnitude than those for the low-order scheme. Actual convergence rates will be compared below in Table 4.5.

Spatial convergence results for coupling method 2, $\mathrm{DG} \leftrightarrow \mathrm{ADCIRC}$, are presented in Figure 4.15. Notice that the unlimited higher-order scheme has smaller errors

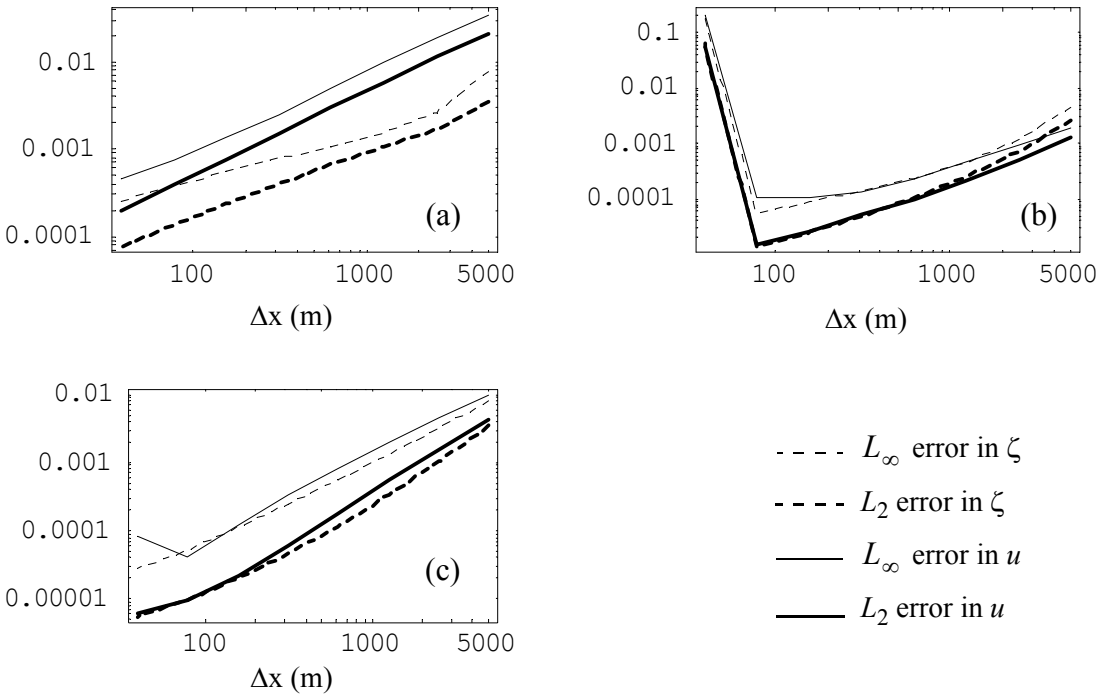


Figure 4.14 Spatial grid convergence for DG/CG model: (a) piecewise constants, (b) piecewise linear with no limiting, (c) piecewise linear with minmod limiter.

than the minmod limited scheme; although, the minmod version has smoother convergence. Meanwhile, the low-order version has higher errors than either of the higher-order versions. Convergence rates derived from the linear portion of the curves are presented below in Table 4.5.

Spatial convergence results for coupling method 3, $\mathrm{DG/CG} \leftrightarrow \mathrm{ADCIRC}$, are presented in Figure 4.16. Notice that the errors for the low-order and minmod schemes are nearly equivalent, while the unlimited higher-order version is unstable. Convergence rates derived from the linear portion of the low-order and minmod curves are presented below in Table 4.5, while no rates are computed for the unlimited scheme, since it is unstable.

Spatial convergence results for all three coupling methods compared with the domain-wide DG and ADCIRC models are presented in Figure 4.17. The results for the L_∞

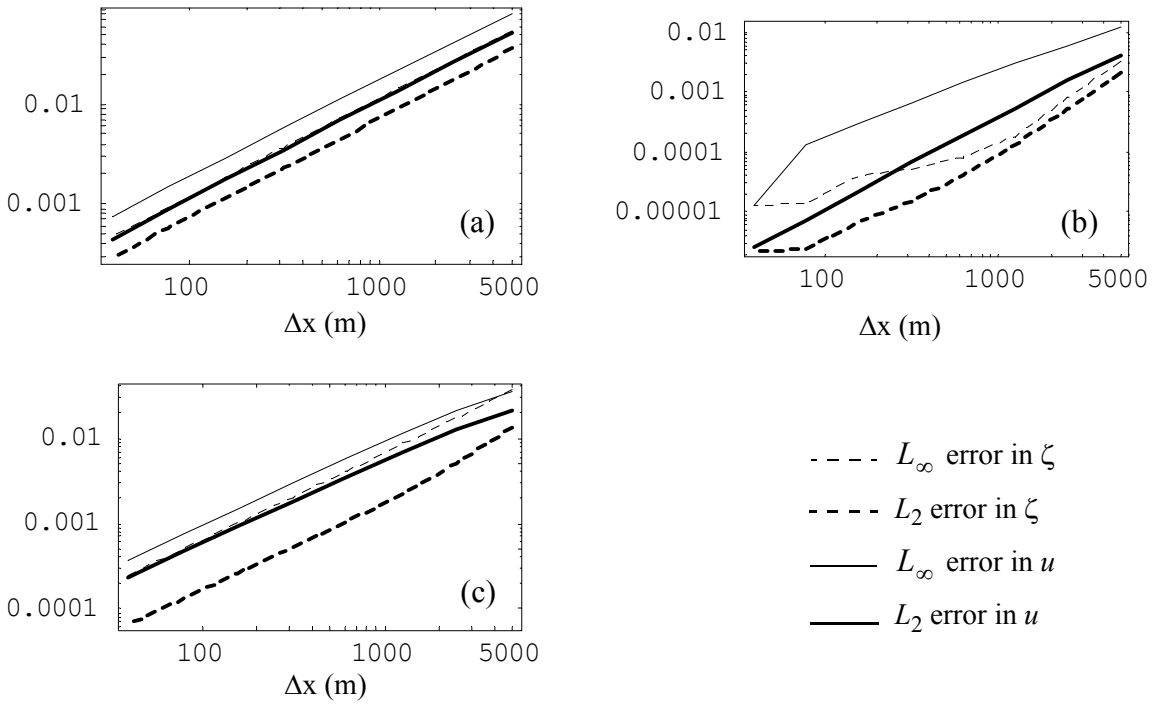


Figure 4.15 Spatial grid convergence for DG <–> ADCIRC model: (a) piecewise constants, (b) piecewise linear with no limiting, (c) piecewise linear with minmod limiter.

error norms are not shown, since they exhibit the same trends as the L_2 error norms. Starting with the solid lines, one notices that the domain-wide DG model with piecewise linear unlimited approximations produces lower errors than either the minmod limited or low-order versions. This model also produces slightly lower errors than the ADCIRC model, although it has much more stringent stability constraints. Additionally, notice that the errors for the minmod limited version are closer to the low-order in both magnitude and rate than they are to the unlimited; the use of limiting stabilizes the algorithm but reduces the accuracy. Note that the convergence curve for the ADCIRC model does not extend to Δx = 39.0625m because this resolution was used as the "true" solution, and thus no error computation is available.

Now, looking at the various coupling schemes in relation to these results, one

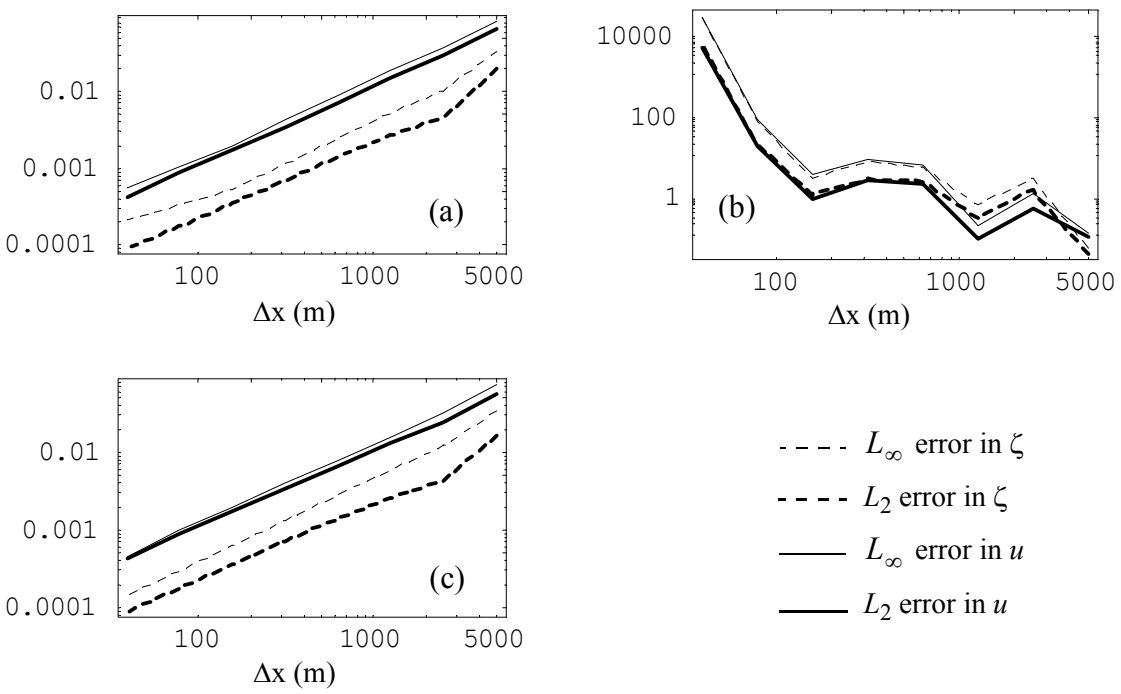


Figure 4.16 Spatial grid convergence for DG/CG <–> ADCIRC model: (a) piecewise constants, (b) piecewise linear with no limiting, (c) piecewise linear with minmod limiter.

notices that the unlimited versions have the lowest errors of the coupling schemes. Additionally, the errors for the unlimited $\mathrm{DG} \rightarrow \mathrm{ADCIRC}$ model are similar to those of the domain-wide ADCIRC model. Otherwise, the higher-order DG/CG versions are the most accurate of the coupled models. Meanwhile, the errors for most of the remaining coupling schemes are larger than those for the domain-wide DG and ADCIRC models, indicating that some accuracy is lost with these coupling schemes. The errors for the two subdomain coupling schemes are similar in magnitude and are higher than all of the other models, with the exception of the domain-wide DG model using piecewise constants or minmod linears, which exhibit higher errors in the elevation data. These higher errors are likely due to the explicit overlapping element used to couple the two subdomains. Although the linear form of the equations used herein did not allow this, flux coupling at the

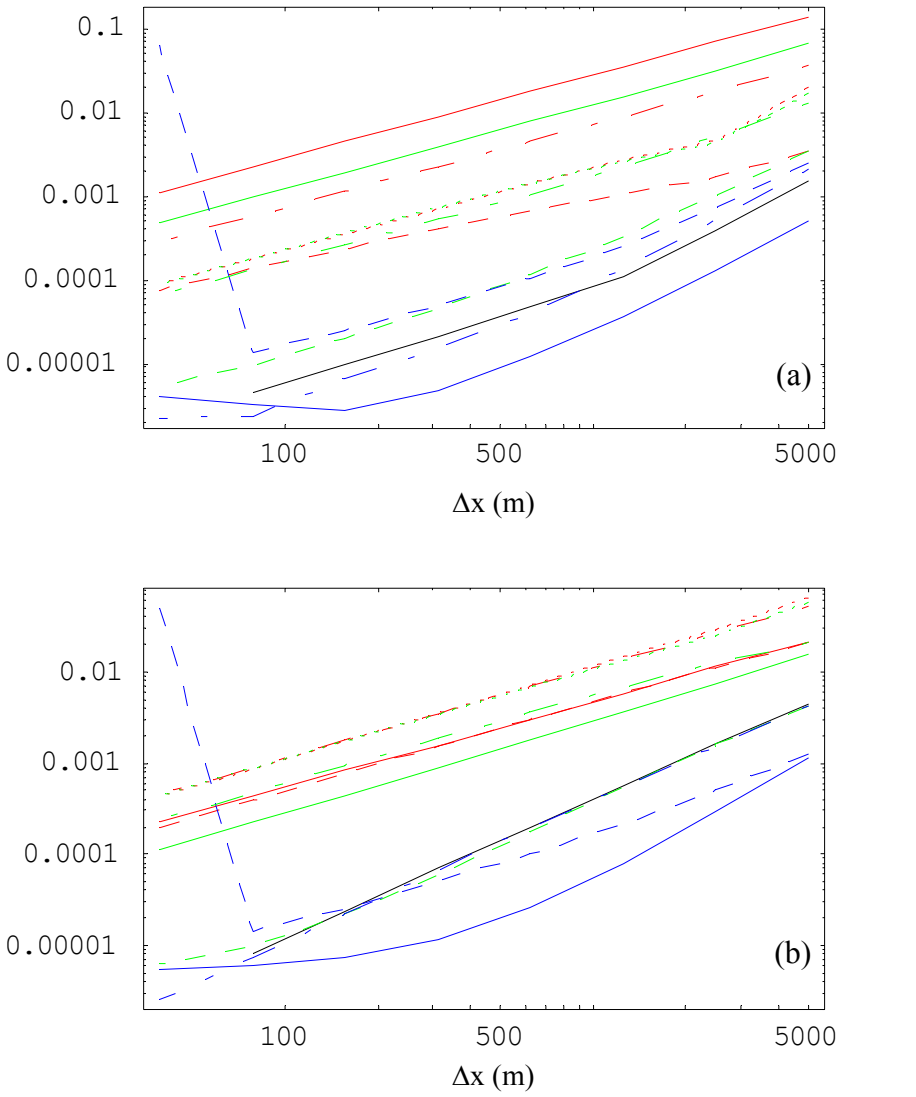


Figure 4.17 Comparison of spatial convergence results for the three coupling models with a domain-wide DG model and the ADCIRC model: (a) L_2 error in ζ, (b) L_2 error in u.

subdomain, as done by *Dawson and Proft* [2002b, 2002c], may be more appropriate. Additionally, in practice the order of the time stepping routine typically matches the order of the spatial approximation, such that for piecewise linear spatial interpolants, a second-order Runge Kutta scheme is used in time. Finally, notice that the errors for the minmod limited DG/CG $\rightarrow$ ADCIRC model are nearly identical to those of the low-order, which indicates that the higher-order versions are unstable, such that the limiter stabilizes the algorithm by setting all of the slopes to zero.

4.7.2 Temporal accuracy

For the temporal convergence study, the test problem and error norms are the same as were used for the spatial convergence. However, instead of using a "true" solution to compare with the various coupled results, interval halving with a base time step of Δt_b = 1.6 seconds was used to evaluate the accuracy of the algorithms, such that the errors are computed for two successive solutions with time steps of $\Delta t_b/m$ and $\Delta t_b/2m$. Each simulation uses a time step one half as large as the previous simulation (interval halving), until the ratio of successive errors converges to a constant. This converged ratio represents the accuracy of the algorithm. In order to isolate the temporal errors, a fixed spatial grid with 640 elements was used for all of the simulations.

A graphic presentation of the results provides for easier comparison between algorithms, therefore only the log-log plots of error versus m will be presented in this section. The tabular data used to generate these plots is available to the interested reader in Appendix E. Also, the average of the converged temporal accuracy for the L_∞ and L_2 error norms for each state variable, elevation and velocity, are reported in Table 4.5 on page 166 within the summary section. Plots for each individual model will be shown first, and then all of the models will be compared.

Temporal convergence results for coupling method 1, equation coupled DG/CG, are presented in Figure 4.18. Notice that the higher-order unlimited approximation is unstable

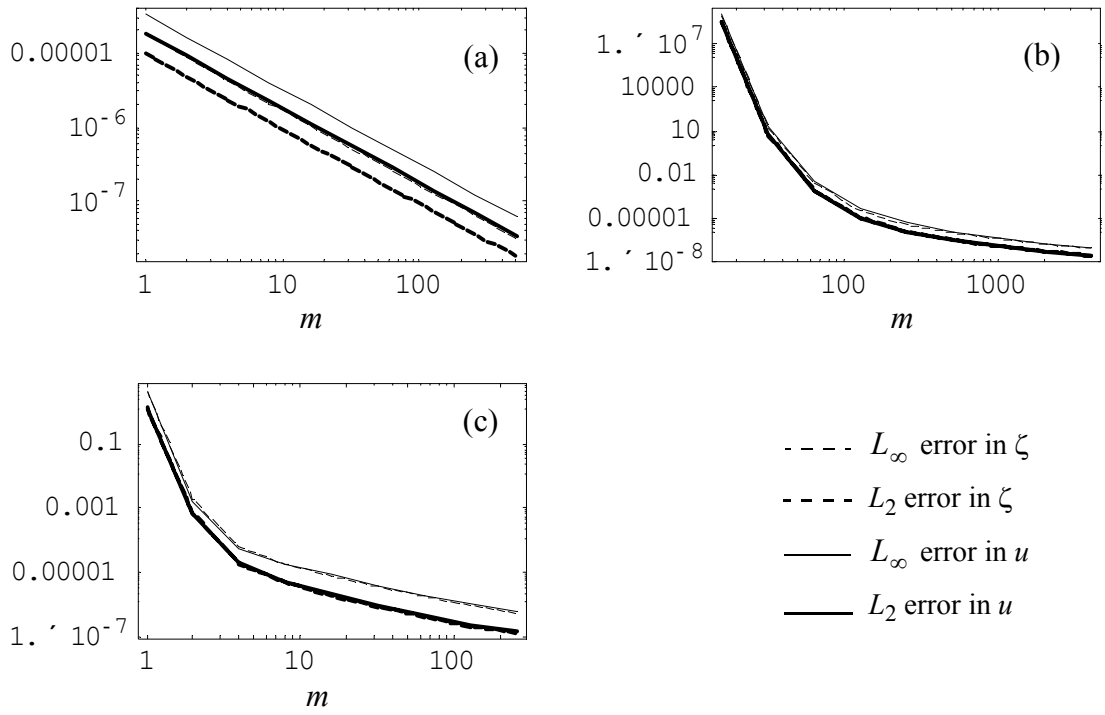


Figure 4.18 Temporal convergence for DG/CG model: (a) piecewise constants, (b) piecewise linear with no limiting, (c) piecewise linear with minmod limiter.

at the larger timesteps, while the minmod limited version remains stable for all time steps. As expected, given the explicit nature of the temporal scheme, all three spatial approximations result in first-order temporal accuracy.

Temporal convergence results for coupling method 2, subdomain coupled DG to ADCIRC, are presented in Figure 4.19. Notice that the unlimited higher-order version is unstable for time steps larger than $\Delta t = 0.2$ seconds ($m \leq 4$), while the minmod limited version remains stable through all time steps. For all three versions, notice that as the time step is decreased past $m = 64$ ($\Delta t = 0.0125$ seconds), the errors begin to increase again due to the boundary conditions in the ADCIRC subdomain. Just before this point, the converged accuracy for all three spatial approximations is around first order.

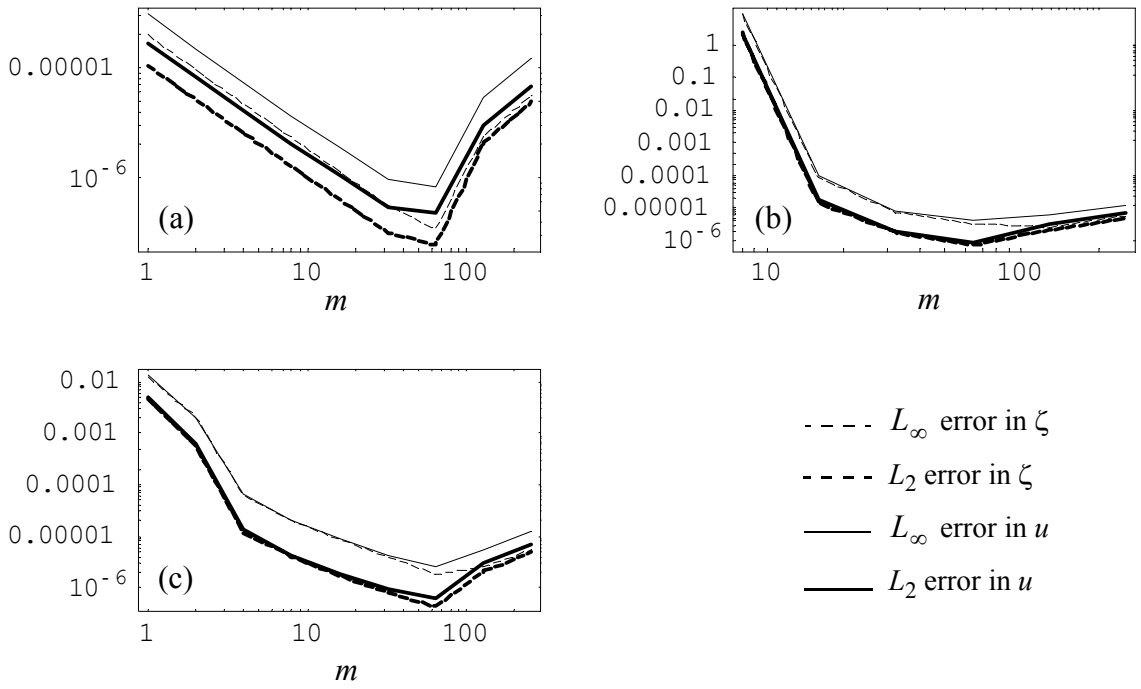


Figure 4.19 Temporal convergence for DG <–> ADCIRC model: (a) piecewise constants, (b) piecewise linear with no limiting, (c) piecewise linear with minmod limiter.

Temporal convergence results for coupling method 3, combined equation and subdomain coupled DG/CG to ADCIRC, are presented in Figure 4.20. Note that the unlimited higher-order version is unstable at all of the time steps that were tried – from 0.8 to 0.0015625 seconds – furthermore, the convergence plot does not indicate that the model will become stable at smaller time steps. Meanwhile, the minmod limited higher-order version remains stable for all time steps, but has higher errors than the low-order version. Both stable versions are first-order accurate in time.

Temporal elevation convergence results for all three coupling methods compared with the domain-wide DG and ADCIRC models are presented in Figure 4.21. The results for the L_∞ error norms are not shown since they exhibit the same behavior as the L_2 error norms. Additionally, the velocity convergence results are nearly identical to the elevation

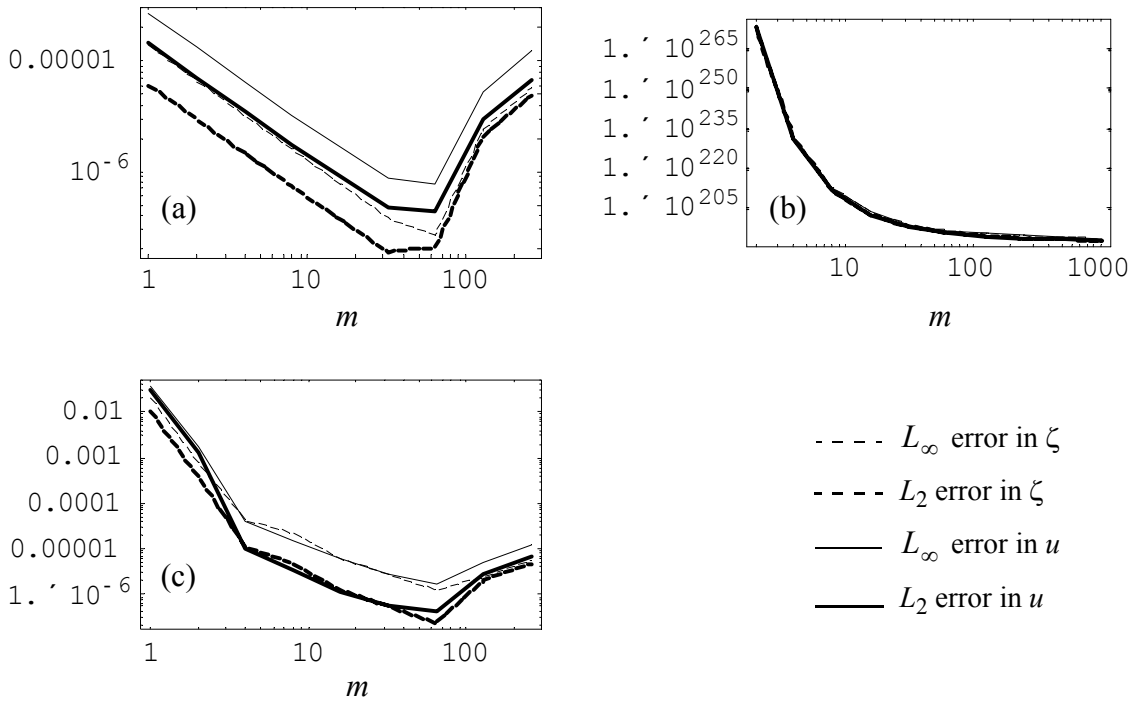


Figure 4.20 Temporal convergence for DG/CG <–> ADCIRC combined coupling scheme: (a) piecewise constants, (b) piecewise linear with no limiting, (c) piecewise linear with minmod limiter.

results, and are not shown herein.

Starting with the solid curves, one notices that the domain-wide ADCIRC model has the lowest temporal errors and one of the highest convergence rates until it begins to diverge at $m = 16$ ($\Delta t = 0.05$ sec.). As noted in Chapter 3, the errors for the ADCIRC model begin to accumulate near the boundaries as the time step is resolved past this point. Meanwhile, the domain-wide DG model with piecewise constant approximations has lower temporal errors than its higher-order counterparts, although the higher-order versions have slightly higher initial convergence rates.

Moving to the coupled models, notice that the $\mathrm{DG} \rightarrow \mathrm{ADCIRC}$ and $\mathrm{DG/CG} \rightarrow \mathrm{ADCIRC}$ models eventually begin to diverge just like the pure ADCIRC

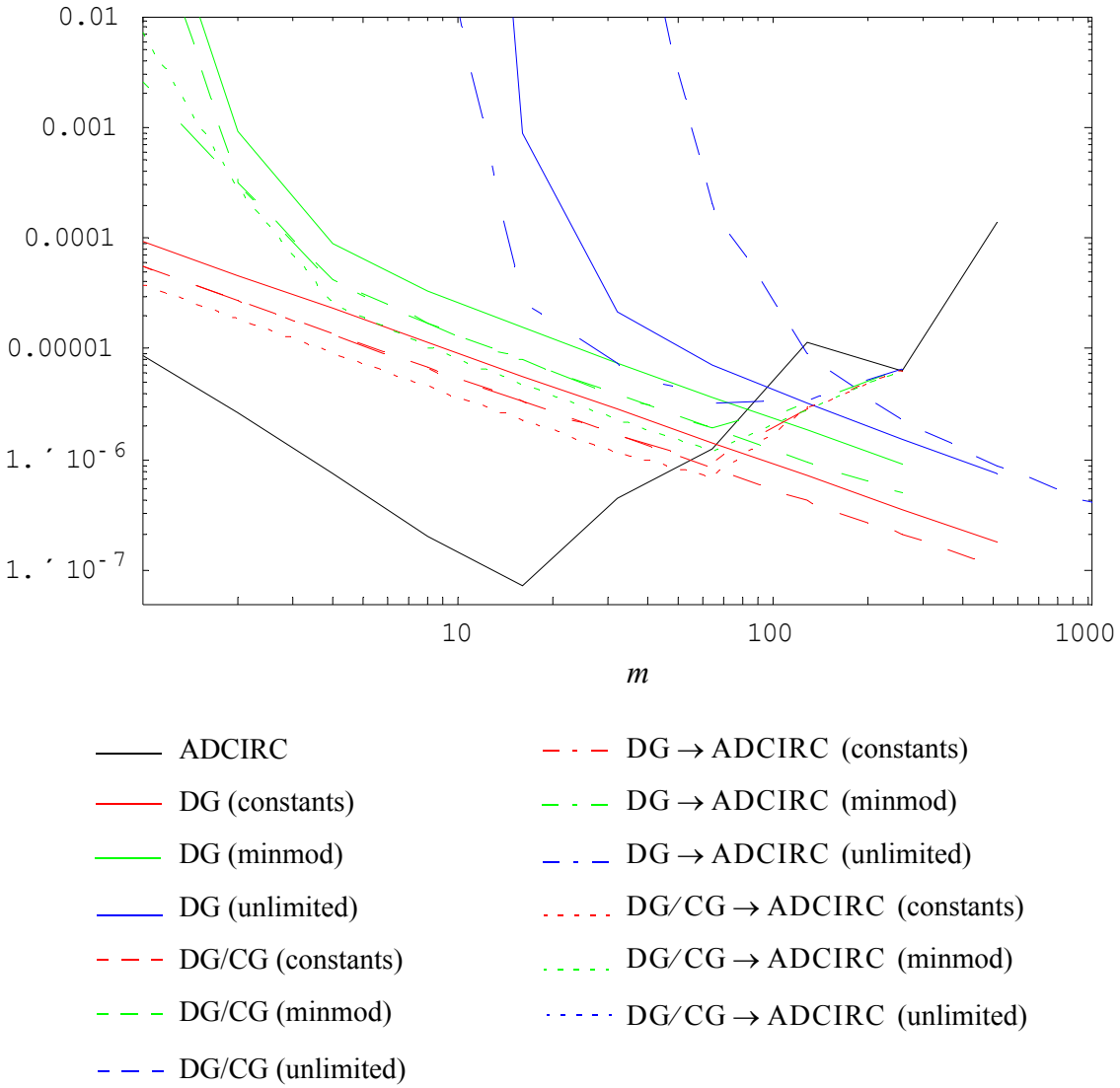


Figure 4.21 Comparison of L_2 temporal convergence results for the elevation state variable for the three coupling models, a domain-wide DG model and the ADCIRC model.

model; although they begin to diverge at a larger m value. Additionally, the error norms for these two coupled models all approach the same diverging solution. Recall also that the unlimited version of the DG/CG → ADCIRC model is unstable at all time steps and is not shown here.

Finally, notice that the low-order versions of the domain-wide DG model and the

various coupling schemes have the lowest temporal errors, followed by the minmod limited versions, and lastly the unlimited versions; with the unlimited DG/CG having the highest temporal errors. The higher-order versions are less accurate than the low-order versions due to the mismatch in the spatial and temporal discretizations. In practice a higher-order Runge-Kutta temporal discretization would be used for the higher-order discontinuous models.

4.7.3 Summary of convergence results

Table 4.5 summarizes the spatial and temporal accuracy of each of the algorithms. The first number in each cell indicates the accuracy of the continuity approximation while the number in parentheses indicates the accuracy of the momentum approximation. These convergence rates are computed from the average of the rates for the two error norms (only the linear range of the spatial convergence data and the converged temporal ratio before any divergence are used), given for the various models in Appendices D and E.

Looking first at the spatial data, one notices that the domain-wide ADCIRC and higher-order DG models have similar convergence rates (between 1.5 and 2.0), while the low-order DG is only first-order accurate. Moving on to the various coupled schemes, one notices that the higher-order versions have somewhat higher convergence rates than the low-order versions; although none of the coupled schemes have convergence rates as high as the domain-wide DG model. However, from Figure 4.17, we recall that coupled model 1 was the only coupled scheme whose errors were of the same magnitude as the domain-wide models. Therefore, the higher-order DG/CG models appear to have the best spatial properties of the coupled models.

Moving now to the temporal convergence, notice that the domain-wide ADCIRC model is second-order accurate, while the temporal accuracy of the various DG and coupled models is roughly first-order. From a strictly temporal point of view, any of the low-order or minmod limited coupled models have acceptable errors. However, in practice a second-

Table 4.5 Accuracy of the coupled continuous and discontinuous algorithms compared to domain-wide ADCIRC and DG models.

Model name	Spatial approximations	Spatial accuracy		Temporal accuracy
		best-fit	peak	
ADCIRC	linears	1.44 (1.02)	1.76 (1.65)	1.96 (2.01)
domain-wide DG	constants	0.99 (0.95)	1.00 (0.98)	1.00 (1.00)
	unlimited	1.51 (1.66)	1.70 (1.80)	1.15 (1.15)
	minmod	1.62 (1.67)	2.04 (1.79)	0.99 (0.99)
Coupled model 1: DG/CG	constants	0.70 (0.94)	1.31 (1.00)	1.00 (1.00)
	unlimited	1.13 (1.02)	1.86 (1.22)	1.17 (1.19)
	minmod	1.26 (1.40)	1.55 (1.58)	0.96 (0.95)
Coupled model 2: DG → ADCIRC	constants	0.99 (0.98)	1.02 (0.99)	1.01 (0.96)
	unlimited	1.41 (1.41)	2.06 (2.44)	1.26 (1.11)
	minmod	1.05 (0.94)	1.22 (0.99)	1.09 (1.07)
Coupled model 3: DG⁄CG → ADCIRC	constants	1.04 (1.03)	1.89 (1.16)	1.01 (0.97)
	unlimited	—	—	—
	minmod	1.06 (1.01)	1.71 (1.19)	1.11 (1.09)

First number is for the continuity approximation and number in parentheses is for the momentum approximation.

order Runge-Kutta time stepping scheme would be used for the higher-order models, so it is difficult to base a recommendation solely on these explicit results.

4.7.4 Comparison of simulation results

The simulation output from the various models for the convergence test case (M_2 tidal forcing on a 50km by 10m channel) are shown in Figure 4.22. The first panel in each row shows the results over the entire domain for the final time (t=90,000 seconds) and the second panel shows the time series at the interface (x=25,000m). A grid spacing of Δx = 78.125m and a time step of Δt = 0.01sec were used to generate these plots. The three domain-wide DG versions (low-order, unlimited linears, minmod linears) plot over the ADCIRC curves (as expected from the error results at Δx = 78.125m in Figure 4.17), so these are not shown explicitly.

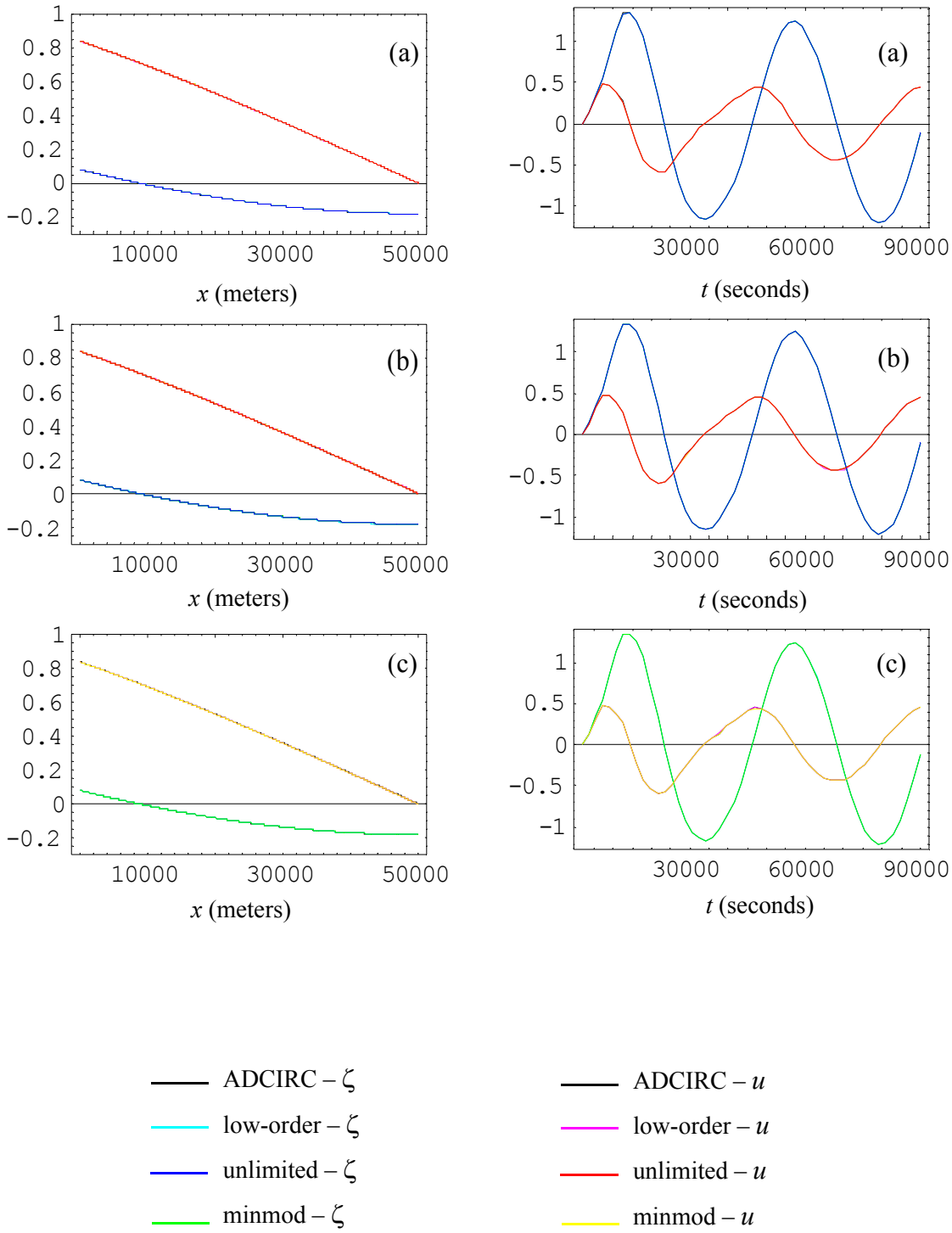


Figure 4.22 Simulation output at $t = 90{,}000$ seconds (left) and time series at $x = 25{,}000$m (right): (a) DG/CG equation coupling, (b) DG <–> ADCIRC subdomain coupling, and (c) DG/CG <–> ADCIRC combined coupling.

Notice that both the domain and time series results for all three coupled models (and spatial approximations) are indistinguishable from the fine-grid ADCIRC results. However, the unlimited version of coupled model 3 is unstable and results are not shown for it. Finally, notice that both of the subdomain coupled models are able to pass the wave through the subdomain interface without distortion (at this fine resolution), as evidenced by the domain and time series plots. The next section will examine the solution quality relative to the location of the subdomain interface for a variable bathymetry test case.

4.8 Qualitative analysis of subdomain interface

This study also seeks to determine the effects of moving the subdomain interface relative to the bathymetric depth. To do this, a one-dimensional slice of the bathymetry off the eastern United States coast is examined, and the quality of the solution monitored as the interface is moved relative to the bathymetry. Figure 4.23 shows the bathymetry for this 1D slice. Notice that this bathymetric profile provides three natural "breaks" in bathymetry – the toe of the slope, the middle where there is a slight change in slope, and the shelf break

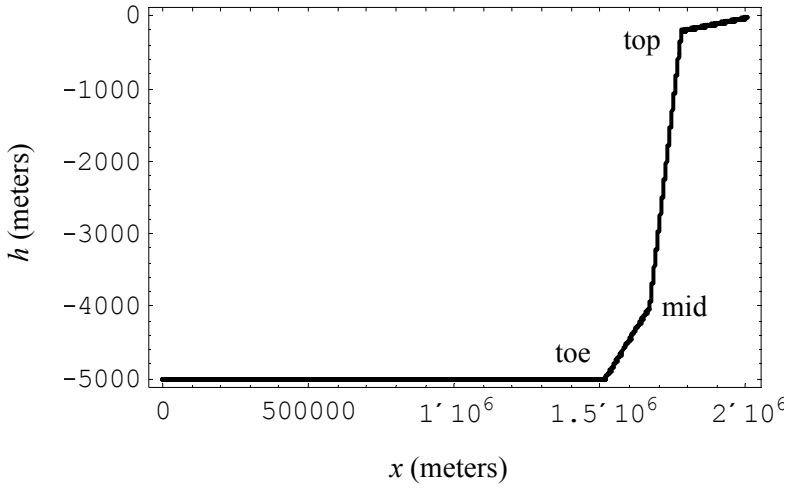


Figure 4.23 Bathymetry profile for subdomain interface tests – 1D slice perpendicular to the eastern United States coast.

at the top of the slope. These natural breaks are indicated in the figure and will serve as the test locations for the subdomain interface.

A fine-grid ADCIRC solution using a constant grid spacing and 8193 nodes is used for the "true" solution. For the comparisons with the coupled models and the domain-wide DG and ADCIRC models, two levels of grid refinement using a constant wavelength to Δx ratio are used: $L/\Delta x = 60$ and $L/\Delta x = 300$. These ratios are typical of those used in practice for coarse and fine grid solutions (see ADCIRC manual, *Luettich et al.* [1992]). The M_2 tidal component is used as the wave of interest, such that the wavelength is calculated as $L = 44712.0\sqrt{gh}$. The grid spacing must change, as the depth varies, in order to keep the ratio constant. The smallest grid spacings will be in the shallow depths where the wavelength is shorter and more refinement is needed. Conversely, the grid spacing is largest in the deep water where the wavelength is longer and less resolution is needed. The specific details for each of these grids are given in Table 4.6 and the bathymetric profiles of each compared to the fine-grid ADCIRC grid are shown in Figure 4.24.

Notice that the bathymetry with a grid ratio of 60 just provides enough resolution

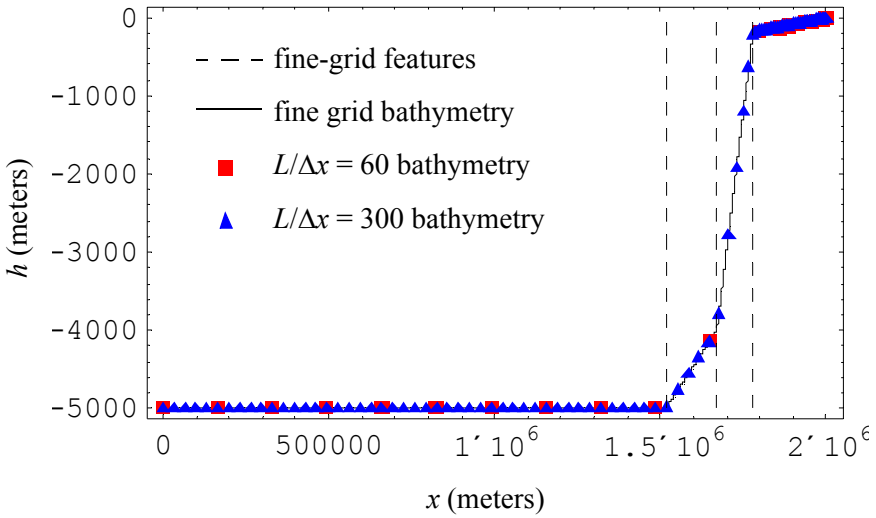


Figure 4.24 Comparison of 1D Eastern United States bathymetry profiles for the three grid resolutions: fine-grid, $L/dx = 60$, $L/dx = 300$.

Table 4.6 Grid details for East coast coupled model simulations.

Grid type	Number of nodes	Range of Δx values (meters)	Location of features, x (meters)		
			toe	mid	top
fine-grid	8193	244.14	1,516,200.0	1,670,200.0	1,779,000.0
$L / \Delta x = 60$	21	12,320.8 to 165,041.2	1,485,370.7	1,650,411.9	1,800,381.5
$L / \Delta x = 300$	108	765.9 to 33,008.2	1,518,379.0	1,676,032.1	1,778,096.5

to capture the features themselves, but does not have any intermediate nodes along the slopes between the toe and top. Meanwhile, the higher ratio of 300 provides good resolution of the slopes themselves while significantly reducing the number of nodes relative to the fine grid. One drawback of a constant $L/\Delta x$ grid is that refining the grid requires the addition of several unnecessary nodes to the flat, deep portion of the domain, as well as the desired nodes along the slopes. Alternate meshing algorithms have been developed that avoid this problem, but $L/\Delta x$ is still commonly-used, so it is chosen for this study.

For each of the three coupled models, as well as the domain-wide ADCIRC and DG models, approximately two periods of the M_2 tide (90,000 seconds) are simulated on each of the coarser grids. A time series of the model output at each of the three test locations – toe, mid, top – are compared against the fine-grid ADCIRC solution. Since the features of the coarser grids do not fall at the same x location, as shown in Table 4.6, the fine-grid results are interpolated to the coarse locations. Recall that coupling method 1 is only an equation coupling scheme, thus there is no subdomain interface. Meanwhile, for coupling methods 2 and 3, results will be shown with the subdomain interface at each of the three locations – toe, middle and top of the slope. Additionally, where appropriate, the discontinuous results are shown at each location. A time step of 0.01 seconds was used for all of the subsequent simulations, including the fine-grid ADCIRC solution.

4.8.1 Simulation results for grid ratio of 60

ADCIRC model

For each of the three bathymetry features, the time series for the domain-wide ADCIRC model at the $L/\Delta x = 60$ resolution are shown in Figure 4.25. Notice that the results with the coarser resolution do a fairly good job of capturing the peaks in the elevation data at the toe and middle of the slope, but do not capture the peaks at the top of the slope. Furthermore, this coarse solution does not capture the peaks in the velocity results at any of the stations. Consequently, one would not expect the various coupled models to match perfectly at this resolution either. The global results for the ADCIRC model are presented with those of the domain-wide DG model below in Figure 4.27.

Domain-wide DG model

The time series for the domain-wide DG model, at each of the three bathymetry features, are shown in Figure 4.26. The plus and minus symbols indicate the discontinuous results from the DG model, where the plus symbol indicates the value to the right of the element boundary and the minus symbol indicates the value to the left.

For the station at the toe, notice that all of the appoximating spaces do an adequate job of capturing the peaks. At the middle station, the right state value for all three approximating spaces captures the peak behavior, while the left state value for the two higher-order approximations deviate significantly from the fine-grid solution. Finally for the station at the top of the slope, the left state value does the best job of capturing the peak behavior. At this station, the right state value for the low-order and unlimited high-order schemes deviate a great deal from the fine-grid solution, while the minmod limiter generally reduces the deviation in left and right state values.

Figure 4.27 compares the model results over the entire domain for the final output at t = 90,000 seconds. The vertical dashed lines indicate the location of the bathymetry features – toe, middle and top of slope – for the grid ratio of 60, as given in Table 4.6. Notice

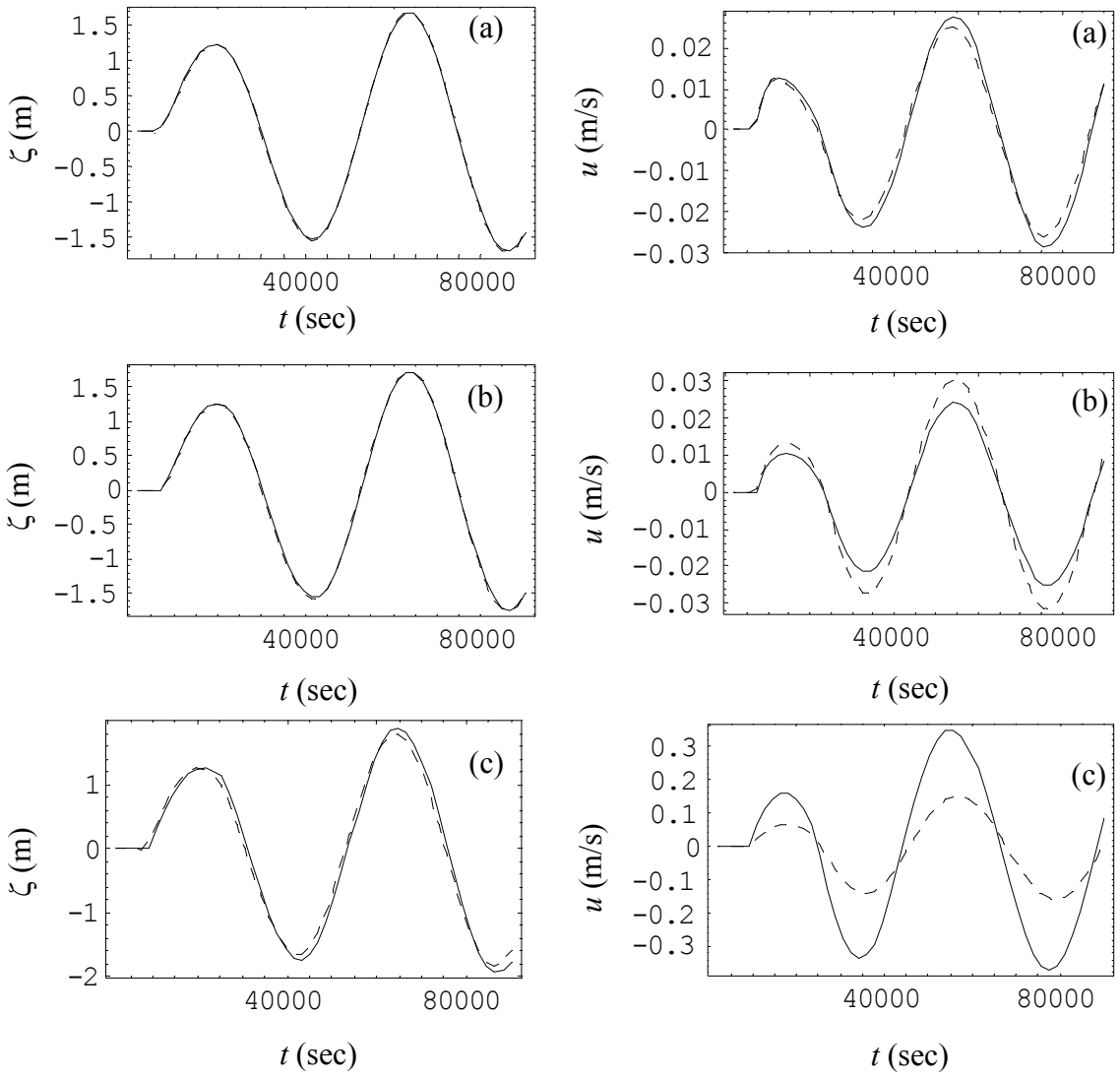


Figure 4.25 Time series at bathymetry stations for domain-wide ADCIRC model simulated on a grid ratio of 60: a) toe of slope, b) middle of slope, c) top of slope.

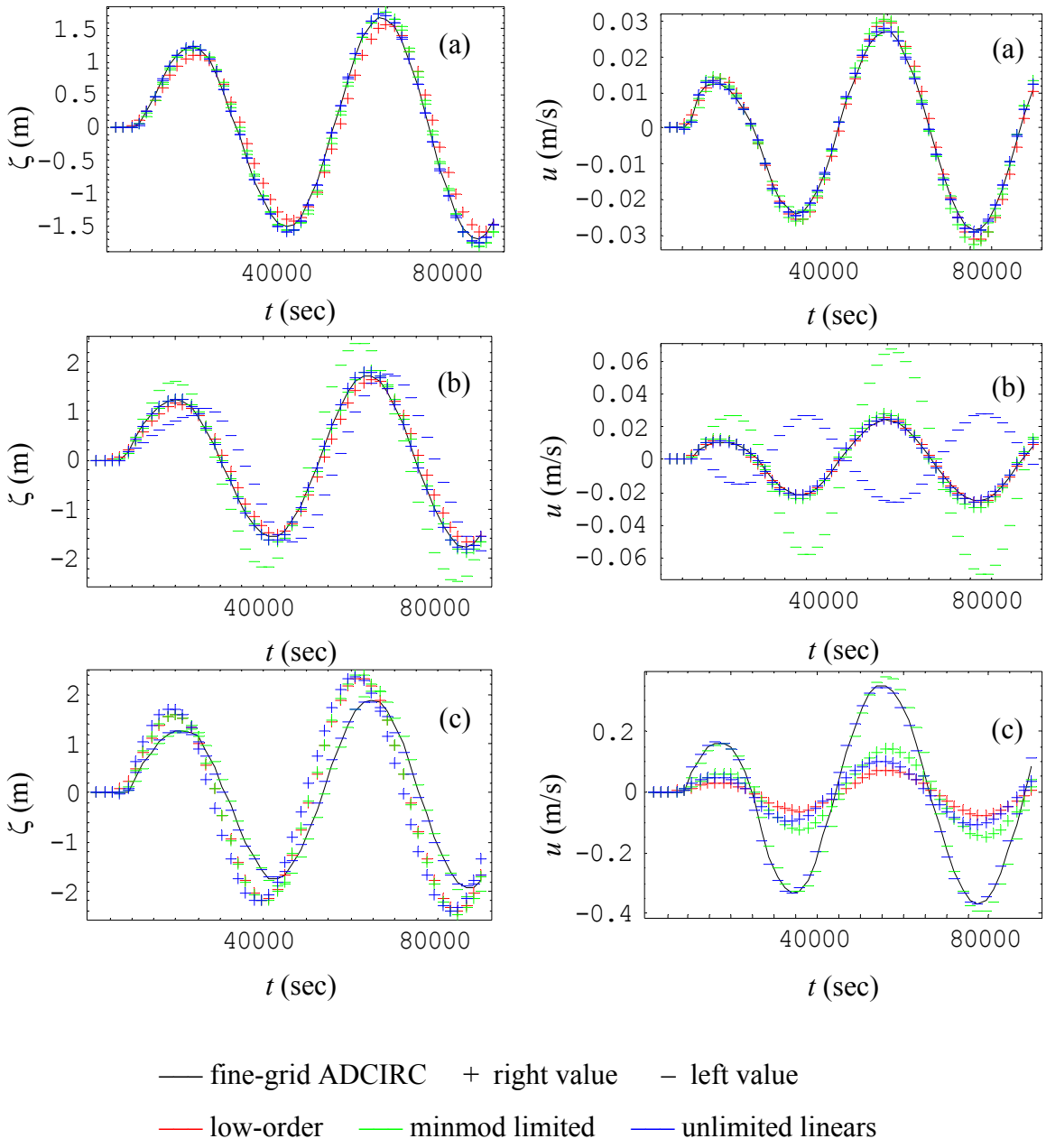


Figure 4.26 Time series at bathymetry stations for domain-wide DG model simulated on a grid ratio of 60: a) toe of slope, b) middle of slope, c) top of slope.

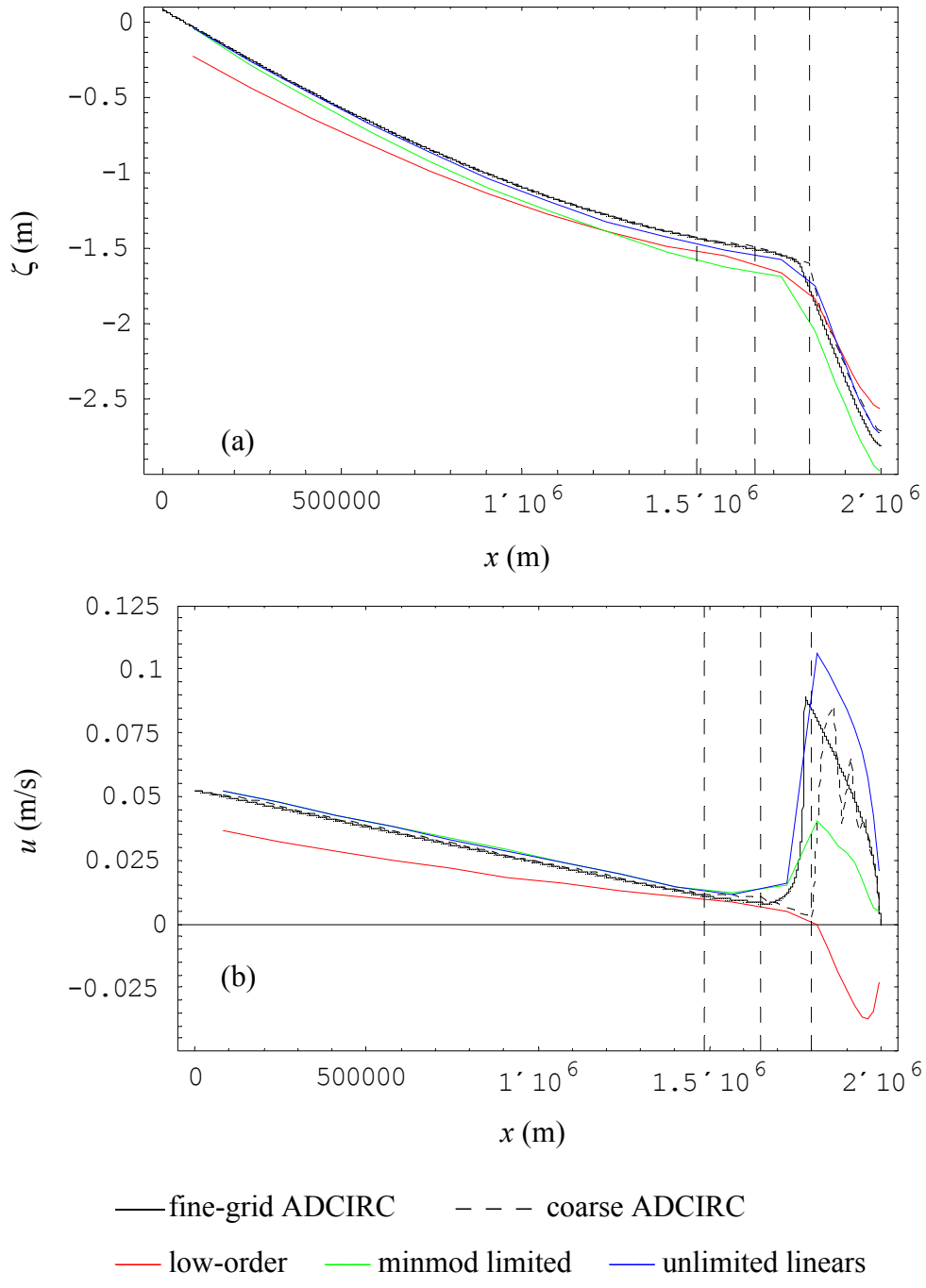


Figure 4.27 Final output at $t = 90{,}000$ seconds for domain-wide DG model simulated on a grid ratio of 60: (a) elevations and (b) velocities.

that none of the DG approximations are able to capture all of the global behavior; although the unlimited higher-order version captures the general behavior. The minmod limited version is slightly better than the low-order version, but is not as good as the unlimited version. Notice also that the coarse ADCIRC solution has some oscillations on the shelf for the velocity solution and lies slightly above the fine-grid elevation solution.

The elevation and velocity errors were computed for each time series by using the most accurate state value (left or right) for all comparisons that involve discontinuous data. For both state variables, the maximum and L_2 average errors over the entire time series were calculated and are summarized for all of the versions in Table 4.7. Additionally, the average and maximum errors over the entire domain are computed for the last temporal output and included in the table, denoted by the word "global" in the station column. The minimum errors for each station and the minimum over the entire domain are indicated by bold-italics in each column.

Notice that the unlimited higher-order version has the lowest errors locally (for all stations) and globally for the domain-wide DG model. Notice also that the coarser ADCIRC model has slightly lower errors for the elevation results; but that the domain-wide DG model has lower errors for the velocity, particularly on the shelf.

Coupled model 1

The time series for coupled model 1, at each of the three bathymetry features, are shown in Figure 4.28. The plus and minus symbols indicate the results from the coupled model, where for discontinuous results the plus symbol indicates the value to the right of the element boundary and the minus symbol indicates the value to the left. The elevations are discontinuous with this coupled model while the velocity results are continuous.

For the station at the toe of the slope, notice that the low-order and minmod limited versions capture the behavior of the elevation and velocity, while the unlimited version overshoots all of the local maxima and minima. The same behavior is noted for the station

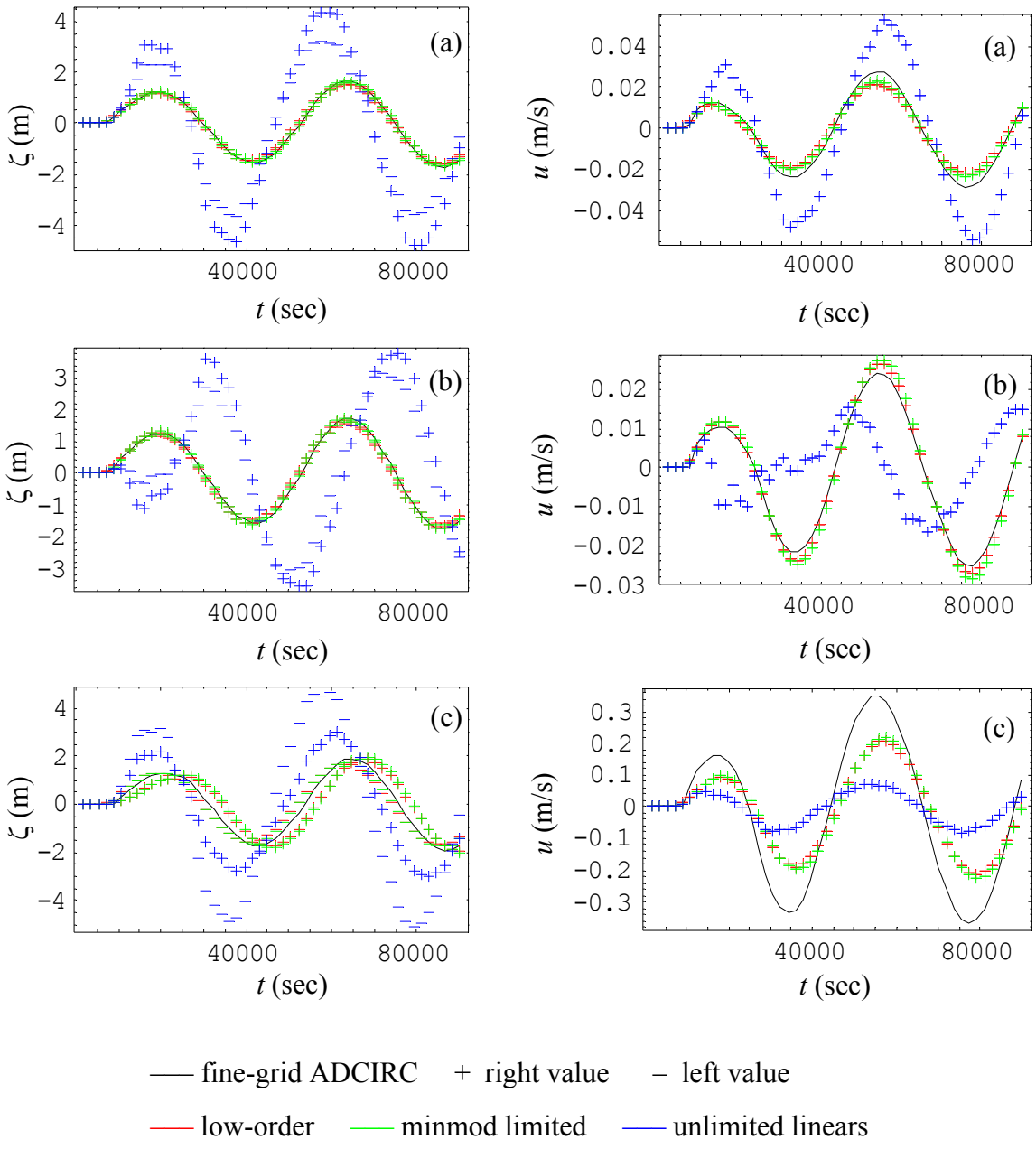


Figure 4.28 Time series at bathymetry stations for DG/CG model simulated on a grid ratio of 60: a) toe of slope, b) middle of slope, c) top of slope.

Table 4.7 Spatial errors for domain-wide models simulated on a grid ratio of 60.

Spatial approximation	Station location	L_2 errors		L_∞ errors	
		elevation	velocity	elevation	velocity
domain-wide ADCIRC model					
continuous linears	toe	***0.0125***	0.0016	***0.0254***	0.0025
	mid	***0.0273***	0.0035	***0.0524***	0.0060
	top	***0.0183***	0.1031	***0.0343***	0.1758
	global	***0.0649***	0.0191	0.1571	0.0814
domain-wide DG model					
piecewise constants	toe	0.1799 ^{R}	0.0018 ^{R}	0.3161 ^{R}	0.0035 ^{R}
	mid	0.1579 ^{L}	0.0010 ^{L}	0.2794 ^{L}	0.0031 ^{L}
	top	0.1799 ^{R}	0.0534 ^{R}	0.3198 ^{R}	0.0918 ^{R}
	global	0.1541	0.0500	0.2421	0.0826
unlimited linears	toe	0.0365 ^{R}	***0.0009*** ^{R}	0.0679 ^{R}	***0.0016*** ^{R}
	mid	0.0404 ^{L}	***0.0010*** ^{L}	0.0733 ^{L}	***0.0020*** ^{L}
	top	0.0879 ^{R}	***0.0186*** ^{R}	0.1576 ^{R}	***0.0353*** ^{R}
	global	0.0659	***0.0167***	***0.1028***	***0.0282***
minmod linears	toe	0.0761 ^{L}	0.0010 ^{R}	0.1474 ^{L}	0.0034 ^{R}
	mid	0.0767 ^{L}	0.0024 ^{L}	0.1529 ^{L}	0.0042 ^{L}
	top	0.1128 ^{R}	0.0322 ^{R}	0.2311 ^{R}	0.0693 ^{R}
	global	0.1393	0.0184	0.2003	0.0394

The bold-italics type face indicates the minimum error for each station and globally. The superscripts denote which discontinuous value was used to compute the errors for the domain-wide DG model: ^{L} left or ^{R} right.

at the middle of the slope, although the velocity solutions are not as good of a match. For the station at the top of the slope, the low-order and minmod limited elevations are correct on average, while the unlimited version overshoots the peaks. Meanwhile, none of the approximations capture the velocity behavior at this station.

Figure 4.29 compares the model results over the entire domain for the final output at t = 90,000 seconds. The vertical dashed lines indicate the location of the bathymetry features – toe, middle and top of slope – for the grid ratio of 60, as given in Table 4.6. Notice that the unlimited approximation oscillates about the true solution, while all three spatial

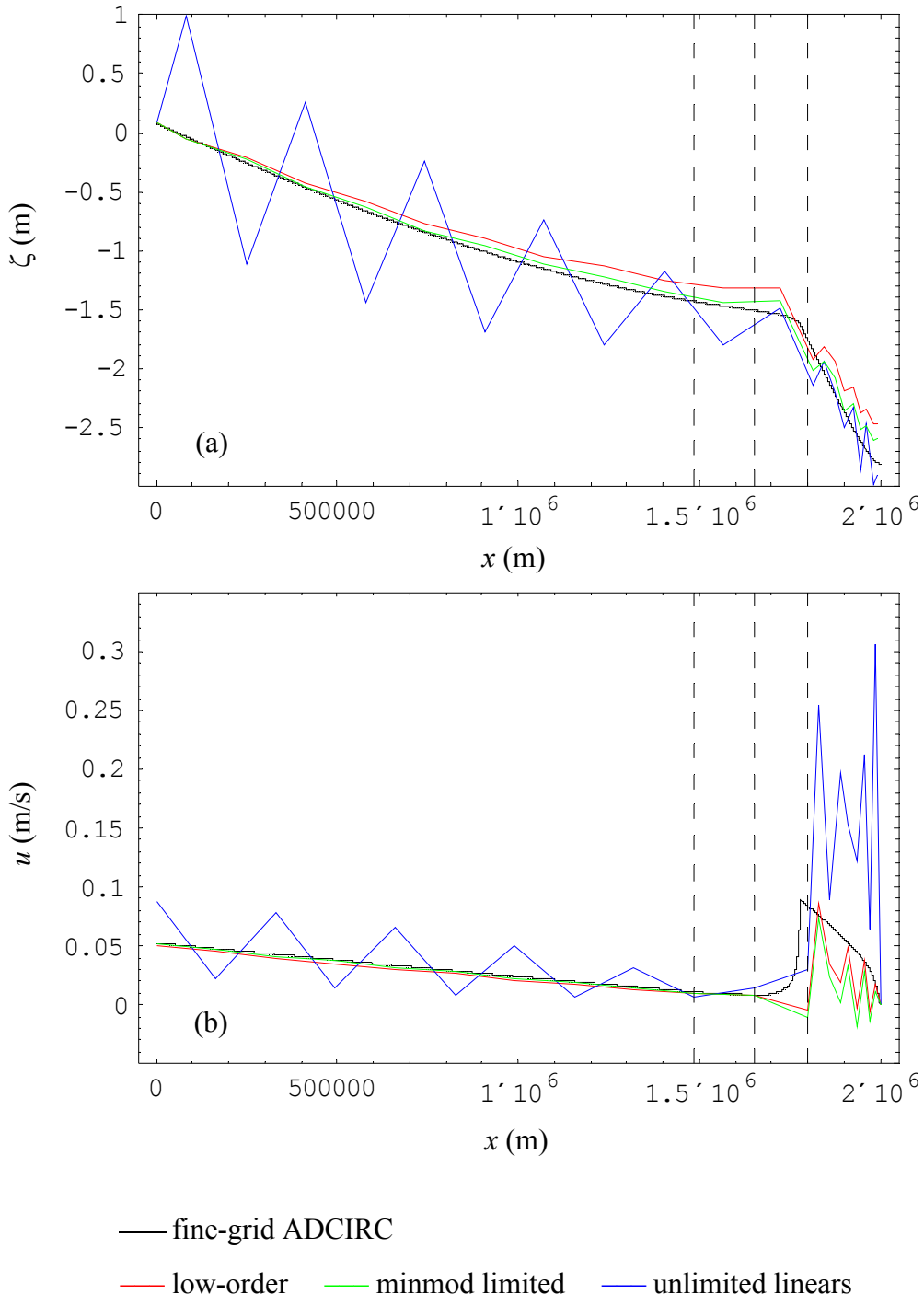


Figure 4.29 Final output at $t = 90{,}000$ seconds for DG/CG model simulated on a grid ratio of 60: (a) elevations and (b) velocities.

approximations have oscillations on the shelf. None of the approximations match the fine-grid solution over the entire domain at this resolution.

The elevation and velocity errors were computed for each time series by using the most accurate state value (left or right) for all comparisons that involve discontinuous data. For both state variables, the maximum and L_2 average errors over the entire time series were calculated and are summarized for all of the versions in Table 4.8. Additionally, the average and maximum errors over the entire domain are computed for the last temporal output and included in the table, denoted by the word "global" in the station column. The minimum errors are indicated by the bold-italics typeface in each column. From these results alone, it is unclear whether the piecewise constant or minmod limited results are best overall. Further comparisons will be done with a finer grid in §4.8.2.

Table 4.8 Spatial errors for coupled model 1 simulated on a grid ratio of 60.

Spatial approximation	Station location	L_2 errors		L_∞ errors	
		elevation	velocity	elevation	velocity
piecewise constants	toe	0.0715 ^{R}	0.0037	0.1335 ^{R}	0.0065
	mid	0.1111 ^{L}	***0.0014***	0.1972 ^{L}	***0.0024***
	top	0.2976 ^{L}	0.1006	0.5043 ^{L}	0.1712
	global	0.1960	***0.0261***	0.3534	***0.0883***
unlimited linears	toe	1.5699 ^{L}	0.0167	2.7873 ^{L}	0.0279
	mid	1.8657 ^{L}	0.0162	3.2193 ^{L}	0.0271
	top	1.1013 ^{R}	0.1722	1.9266 ^{R}	0.2927
	global	0.4697	0.0956	1.0257	0.2887
minmod linears	toe	***0.0467*** ^{R}	***0.0028***	***0.0943*** ^{R}	***0.0053***
	mid	***0.0590*** ^{L}	0.0023	***0.1129*** ^{L}	0.0040
	top	***0.2396*** ^{L}	***0.0985***	***0.4141*** ^{L}	***0.1658***
	global	***0.1094***	0.0313	***0.2236***	0.0943

The superscripts indicate which discontinuous state was used to compute the errors: ^{L} left or ^{R} right.
The minimum errors are denoted by bold-italics.

Coupled model 2

As with the previous model, the plus and minus symbols in the following three figures indicate the results from the coupled model, where for discontinuous results the plus symbol indicates the state value to the right of the element boundary and the minus symbol indicates the value to the left. Both the elevations and velocities are discontinuous in the DG subdomain, while both variables in the ADCIRC subdomain are continuous. Three figures will be presented for this model, one for each location of the subdomain interface.

The time series for coupled model 2 with the subdomain interface located at the toe of the slope are shown in Figure 4.30. Since the bathymetry features are to the right of the subdomain interface, all of the time series are located in the DG subdomain and are discontinuous.

For the station at the toe of the slope, where the interface is located, the higher-order approximations do a better job of capturing the elevation behavior than the low-order version. However, the right state value from the low-order solution follows the general trends. For the velocity data, the left state values from the unlimited higher-order approximation and the right state values from the low-order approximation do the best job of capturing the local maxima and minima. For the station at the middle of the slope, the left state from all three spatial approximations is able to capture the elevation behavior. The right state value from all three approximations overshoot the peaks for the velocity at this station, while the left states for the low-order and minmod limited versions are able to capture the correct behavior. For the station at the top of the slope, the right state values of all three approximations come closest to capturing the correct behavior for both elevation and velocity.

The time series for coupled model 2 with the subdomain interface located at the middle of the slope are shown in Figure 4.31. The time series at the toe is continuous since it lies in the ADCIRC subdomain and the other two time series are discontinuous.

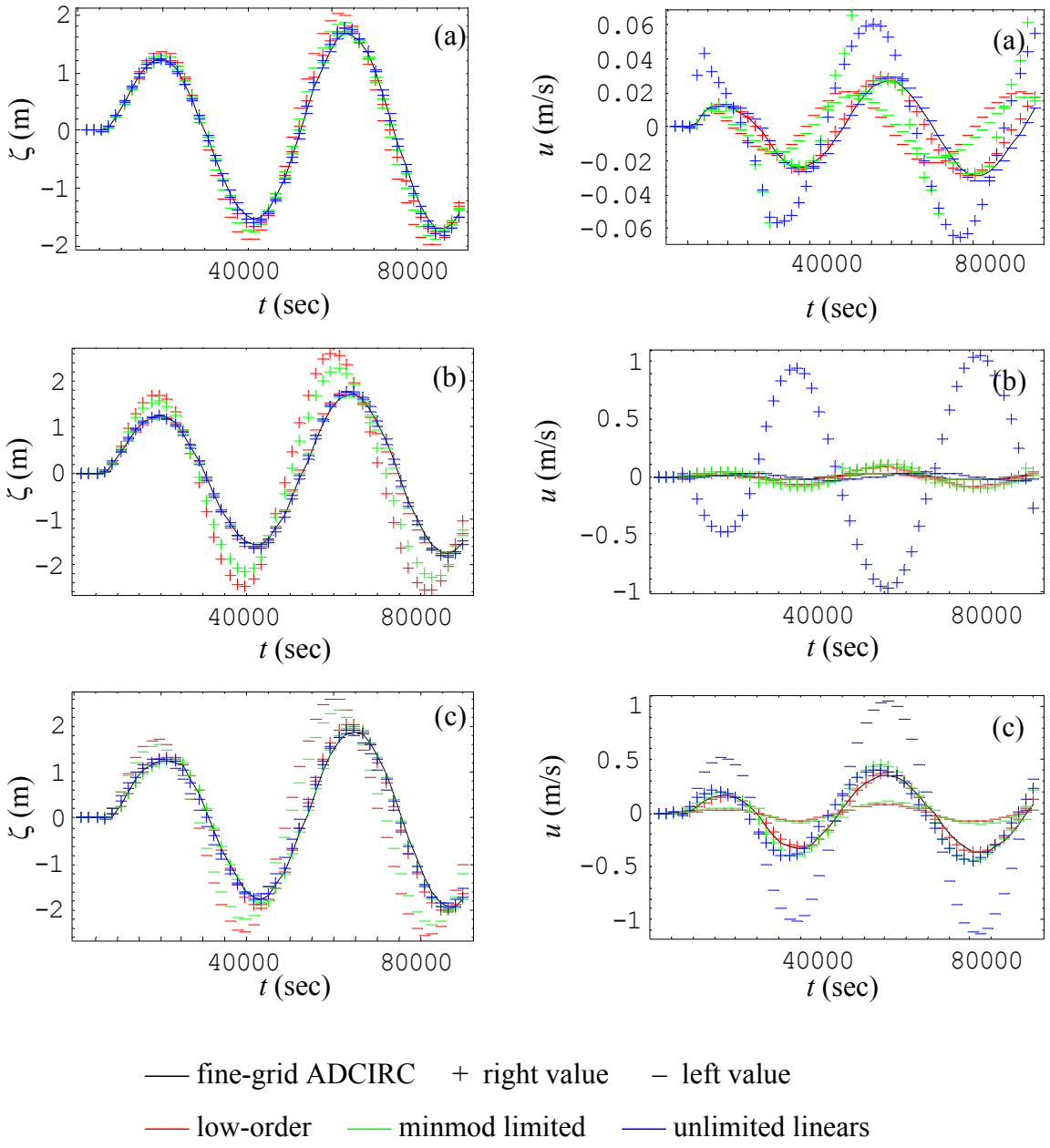


Figure 4.30 Time series at bathymetry stations for DG <–> ADCIRC model with the subdomain interface located at the toe of the slope and simulated on a grid ratio of 60: a) toe of slope, b) middle of slope, c) top of slope.

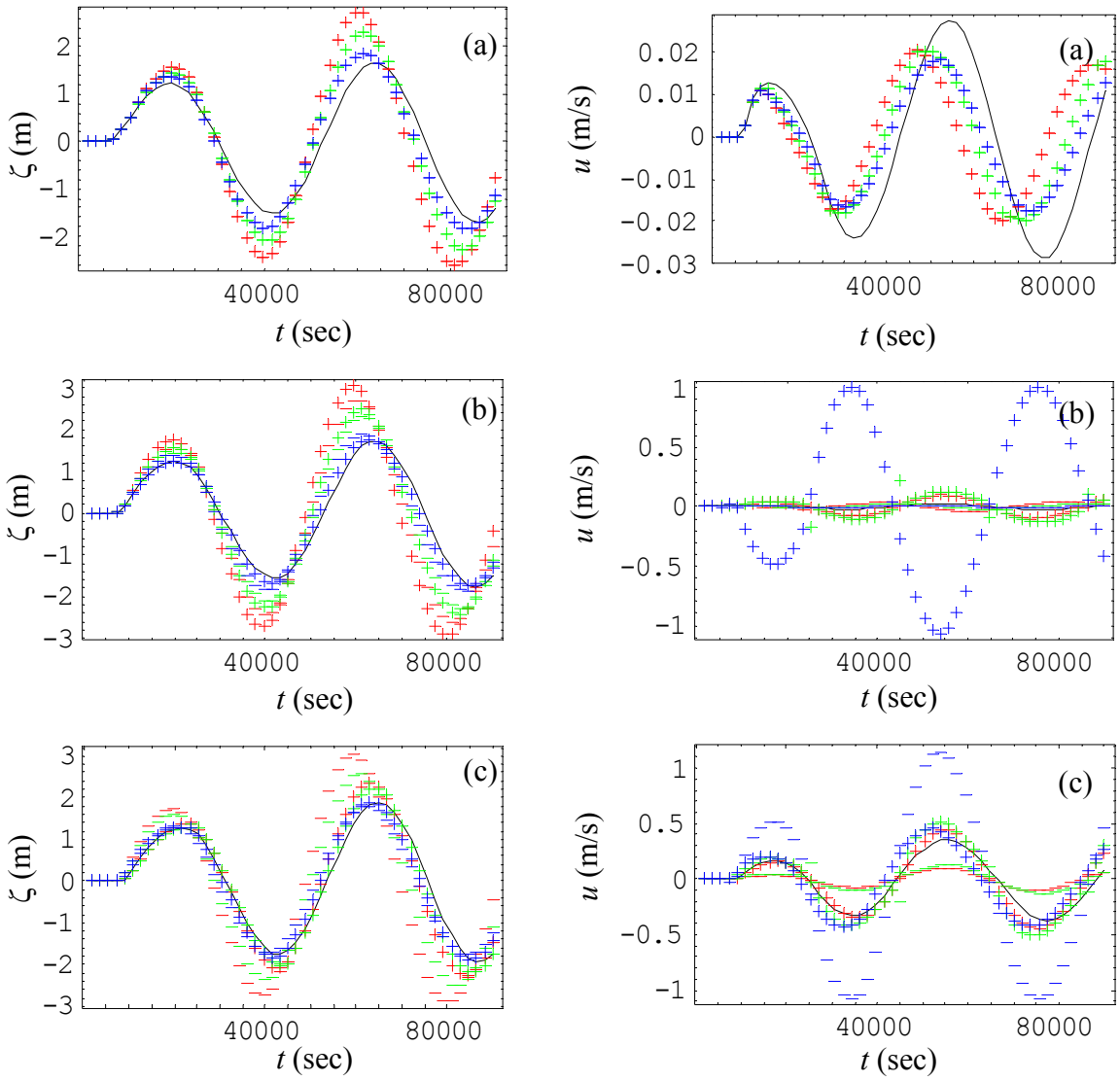


Figure 4.31 Time series at bathymetry stations for DG <–> ADCIRC model with the subdomain interface located at the middle of the slope; simulated on a grid ratio of 60: a) toe of slope, b) middle of slope, c) top of slope.

For the station at the toe of the slope, all of the data is continuous and the unlimited higher-order approximation does the best job of capturing the elevation behavior, while none of the approximations are able to capture the velocity behavior. For the station at the middle of the slope, the low-order and minmod approximations overshoot the peaks for the elevation, while the unlimited higher-order version captures the general trends. Meanwhile, the left state of the minmod limited version comes closest to matching the velocity behavior. Finally, for the station at the top of the slope, the right state values for all three approximations capture the general behavior of the elevation and velocity, with the unlimited version being the closest; but none of the three are a match at all of the peaks.

The time series for coupled model 2 with the subdomain interface located at the top of the slope are shown in Figure 4.32. The time series for the toe and middle of the slope are continuous since they are located in the ADCIRC subdomain and the time series at the top is discontinuous. None of the approximating spaces are able to capture the peaks at any of the stations when the interface is located at the top of the slope; however, the general elevation and velocity behavior is captured by any of the approximations.

Global elevation and velocity results for all three interface locations are presented in Figure 4.33. The vertical dashed lines indicate the location of the bathymetry features – toe, middle and top of slope – for the grid ratio of 60, as given in Table 4.6. Notice that as the subdomain interface moves up the slope and away from the toe, the quality of the global solution degrades quickly. Although, it was not as obvious, this is also evident in the station output in Figures 4.30 through 4.32.

To confirm this, the elevation and velocity errors were computed for each time series by using the most accurate value (left or right) for all comparisons that involve discontinuous data. For both state variables, the maximum and L_2 average errors over the entire time series were calculated for all of the spatial approximations and interface locations. The station errors are summarized in Table 4.9. Additionally, the average and

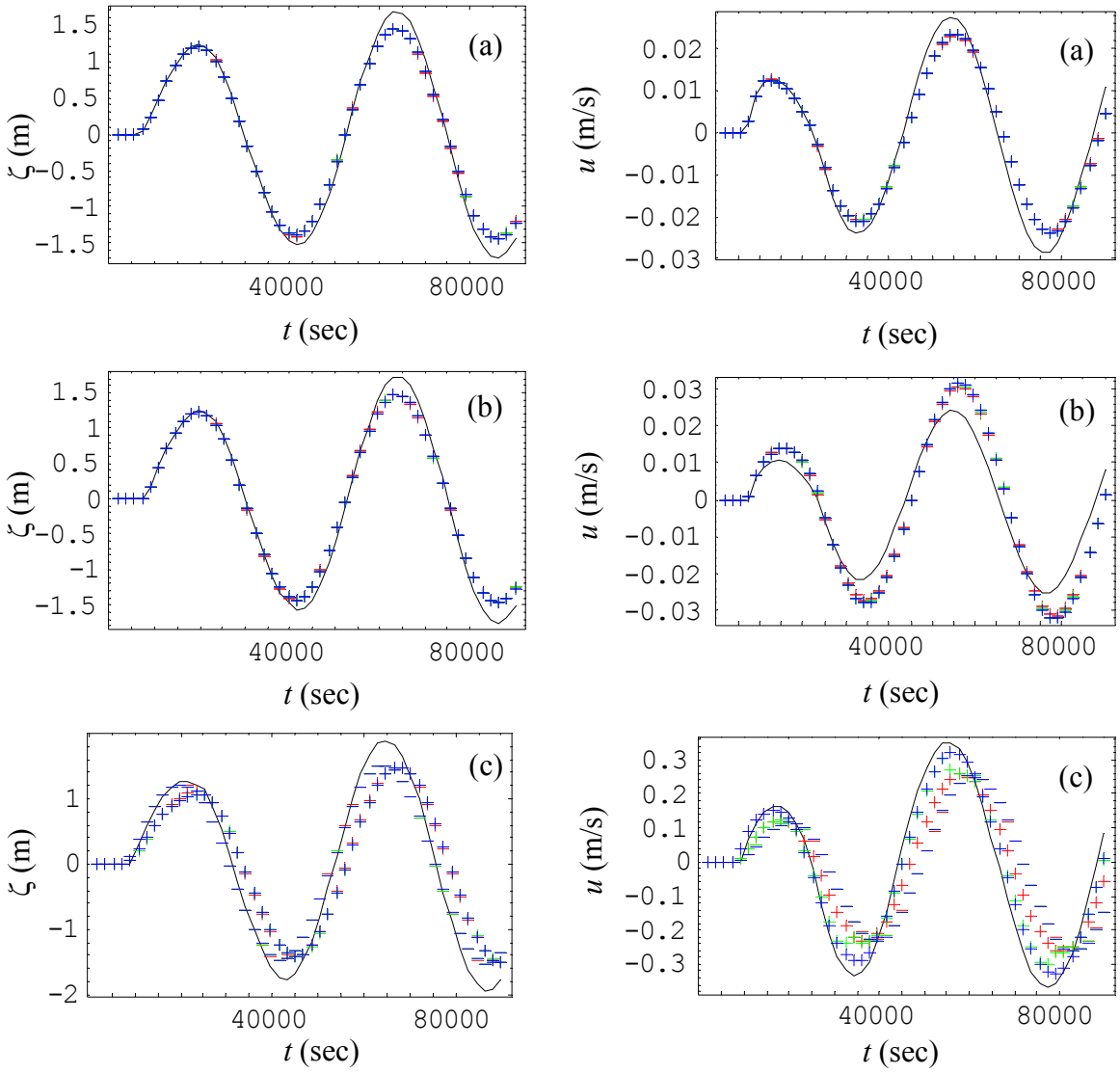


Figure 4.32 Time series at bathymetry stations for DG <–> ADCIRC model with the subdomain interface located at the top of the slope and simulated on a grid ratio of 60: a) toe of slope, b) middle of slope, c) top of slope.

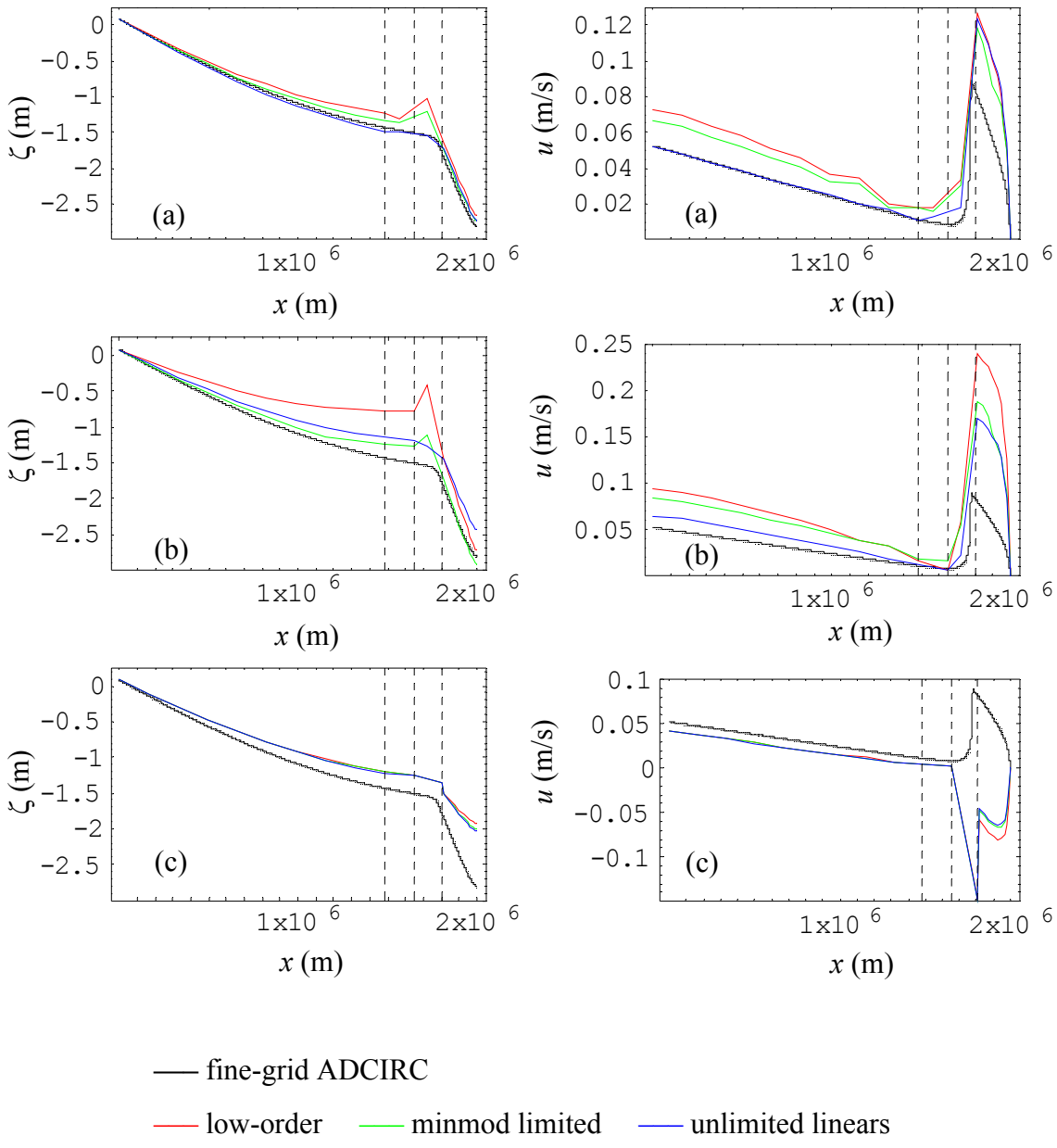


Figure 4.33 Final output at t = 90,000 seconds for DG <–> ADCIRC model simulated on a grid ratio of 60 with the subdomain interface located at the: (a) toe, (b) middle, and (c) top.

Table 4.9 Station errors for DG <–> ADCIRC model simulated on a grid ratio of 60.

Spatial approximation	Interface location	Station location	L_2 errors		L_∞ errors	
			elevation	velocity	elevation	velocity
piecewise constants	toe	toe	0.1195 ^{R}	0.0041 ^{R}	0.2491 ^{R}	0.0085 ^{R}
		mid	0.1371 ^{L}	***0.0055*** ^{L}	0.2984 ^{L}	0.0101 ^{L}
		top	0.1584 ^{R}	***0.0241*** ^{R}	0.3412 ^{R}	***0.0501*** ^{R}
unlimited linears	toe	toe	***0.0395*** ^{L}	***0.0021*** ^{L}	***0.0700*** ^{L}	***0.0033*** ^{L}
		mid	***0.0444*** ^{R}	0.0249 ^{L}	***0.0811*** ^{R}	0.0396 ^{L}
		top	0.0987 ^{R}	0.0922 ^{R}	0.1708 ^{R}	0.1653 ^{R}
minmod linears	toe	toe	0.0631 ^{R}	0.0168 ^{L}	0.1362 ^{R}	0.0297 ^{L}
		mid	0.0892 ^{L}	0.0171 ^{L}	0.1900 ^{L}	0.0485 ^{L}
		top	***0.0786*** ^{R}	0.0661 ^{R}	***0.1405*** ^{R}	0.1405 ^{R}
piecewise constants	mid	toe	0.8075	0.0167	1.6110	0.0321
		mid	0.8144 ^{L}	0.0354 ^{L}	1.6275 ^{L}	0.0653 ^{L}
		top	0.3649 ^{R}	0.0673 ^{R}	0.8280 ^{R}	0.1548 ^{R}
unlimited linears	mid	toe	0.3193	0.0080	0.5839	0.0146
		mid	0.1430 ^{R}	0.0113 ^{L}	0.2968 ^{R}	0.0198 ^{L}
		top	0.2349 ^{R}	0.1242 ^{R}	0.4802 ^{R}	0.2070 ^{R}
minmod linears	mid	toe	0.4878	0.0108	0.9452	0.0206
		mid	0.4945 ^{L}	0.0215 ^{L}	0.9714 ^{L}	0.0388 ^{L}
		top	0.2124 ^{R}	0.1009 ^{R}	0.4516 ^{R}	0.2188 ^{R}
piecewise constants	top	toe	0.1306	0.0035	0.2601	0.0068
		mid	0.1424	0.0056	0.2795	***0.0099***
		top	0.2295 ^{L}	0.0941 ^{R}	0.4246 ^{L}	0.1645 ^{R}
unlimited linears	top	toe	0.1324	0.0034	0.2613	0.0067
		mid	0.1432	0.0060	0.2803	0.0105
		top	0.2272 ^{L}	0.3089 ^{R}	0.4171 ^{L}	0.0743 ^{R}
minmod linears	top	toe	0.1320	0.0034	0.2616	0.0066
		mid	0.1431	0.0059	0.2811	0.0104
		top	0.2277 ^{L}	0.0536 ^{R}	0.4189 ^{L}	0.1097 ^{R}

The superscripts indicate which state value was used to compute the errors for the discontinuous data: L - left or R - right.
The minimum errors for each station location are indicated by the bold-italics type.

maximum errors over the entire domain were calculated and are summarized for the last temporal output in Table 4.10. In each table, the minimum errors for each station or over the entire domain are indicated by bold-italics in each column.

Table 4.10 Global errors for DG <–> ADCIRC model simulated on a grid ratio of 60.

Spatial approximation	Interface location	L_2 errors		L_∞ errors	
		elevation	velocity	elevation	velocity
piecewise constants	toe	0.3283	0.0602	0.8634	0.1051
	mid	0.6431	0.1416	1.6073	0.2487
	top	0.5623	0.0956	0.9979	0.2328
unlimited linears	toe	***0.0752***	***0.0214***	***0.1152***	***0.0379***
	mid	0.3307	0.0558	0.4393	0.0991
	top	0.5148	0.0882	0.8854	0.2345
minmod linears	toe	0.0969	0.0520	0.2864	0.1430
	mid	0.3038	0.0572	0.5869	0.1209
	top	0.5119	0.0875	0.8835	0.2349

This table summarizes the global errors for the last output at t = 90,000 seconds.
The minimum errors for each column are indicated by bold-italics.

From the graphical output, the station errors and the global errors, the unlimited version with the interface at the toe appears to provide the best match to the fine-grid ADCIRC results.

Coupled model 3

As for the previous models, the plus and minus symbols indicate the results from the coupled model, where the plus symbol indicates the state value to the right of the element boundary and the minus symbol indicates the value to the left. The continuous colored lines indicate the average of the discontinuous data from the domain-wide DG model. Only the elevations are discontinuous in the DG/CG subdomain. Meanwhile, both the elevation and velocity results in the ADCIRC subdomain are continuous and the velocity over the entire domain is discontinuous only at the subdomain interface. Three

figures will be presented for this model, one for each location of the subdomain interface.

The time series for coupled model 3 with the subdomain interface located at the toe of the slope are shown in Figure 4.34. Since the bathymetry features are to the right of the subdomain interface, all of the elevation time series are located in the DG subdomain and are discontinuous. The unlimited linear version of the coupled model is unstable when the interface is located at the toe, so no results are shown for this version.

Both of the remaining stable approximations completely miss the velocity behavior at all three stations. Meanwhile, for the elevation behavior, the coupled solutions start out matching the fine-grid time series and begin to deviate significantly as time progresses. Additionally, as you move from the station at the toe of the slope to the top of the slope, the differences in discontinuous state values (left and right) grow considerably.

The time series for coupled model 3 with the subdomain interface located at the middle of the slope are shown in Figure 4.35. The elevation time series at the toe is continuous, since it lies in the ADCIRC subdomain and the other two elevation series are discontinuous. Again, the unlimited linear version of the coupled model is unstable when the interface is located at the middle of the slope, so no results are shown for this version.

Starting with the station at the toe of the slope, neither approximation captures the elevation or velocity behavior, although they are closer than when the subdomain interface was located at the toe. For the station at the middle of the slope, neither approximation captures the elevation behavior; but the left states, which are computed in the ADCIRC subdomain, are able to capture the velocity behavior. Again at the top of the slope, neither approximation matches the fine-grid behavior. As noted above, the errors become larger as time progresses.

The time series for coupled model 3 with the subdomain interface located at the top of the slope are shown in Figure 4.36. The elevation series for the toe and middle of the slope are continuous, since they are located in the ADCIRC subdomain and the elevation

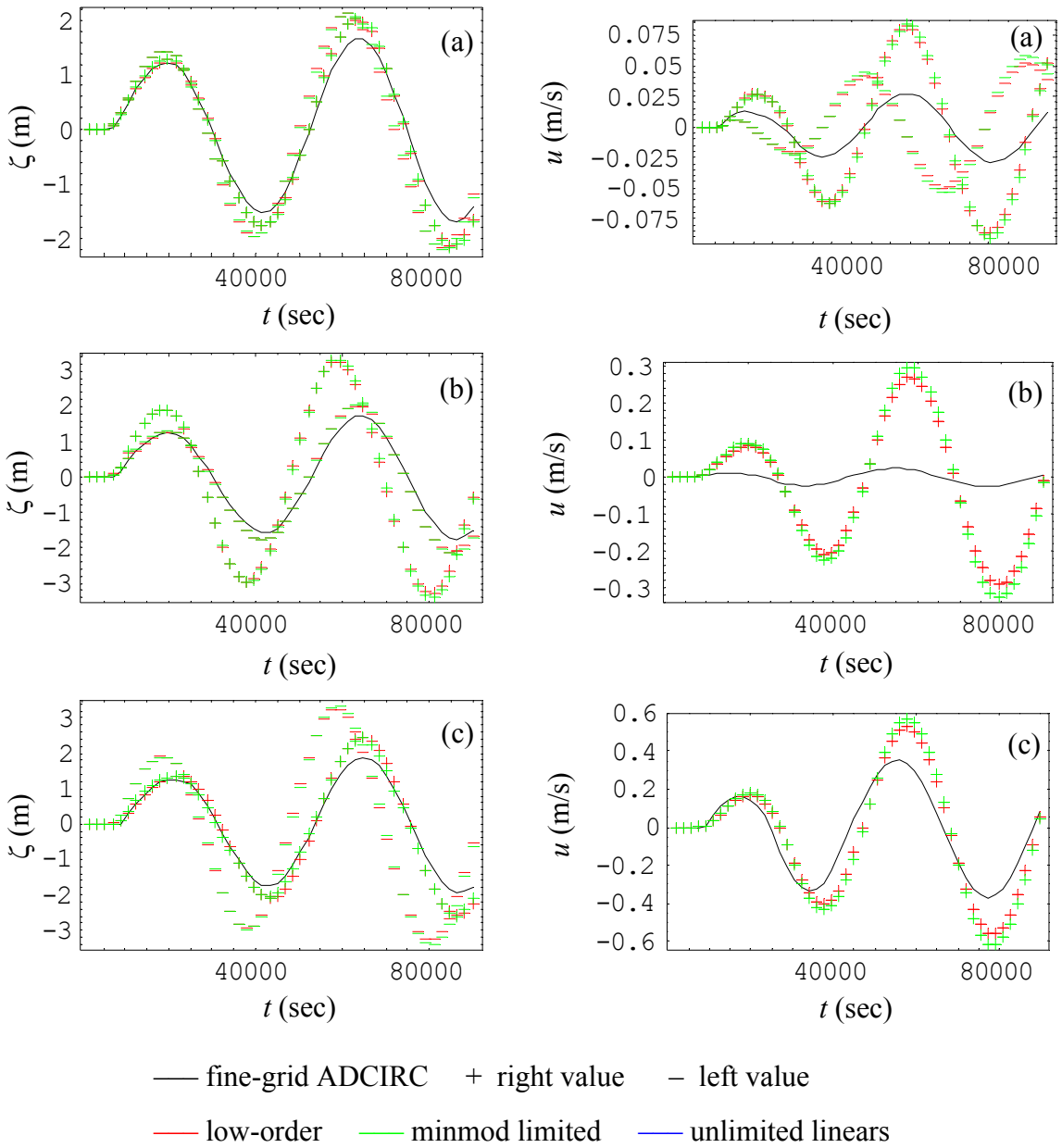


Figure 4.34 Time series at bathymetry stations for DG/CG <–> ADCIRC model with the subdomain interface located at the toe of the slope and simulated on a grid ratio of 60: a) toe of slope, b) middle of slope, c) top of slope.

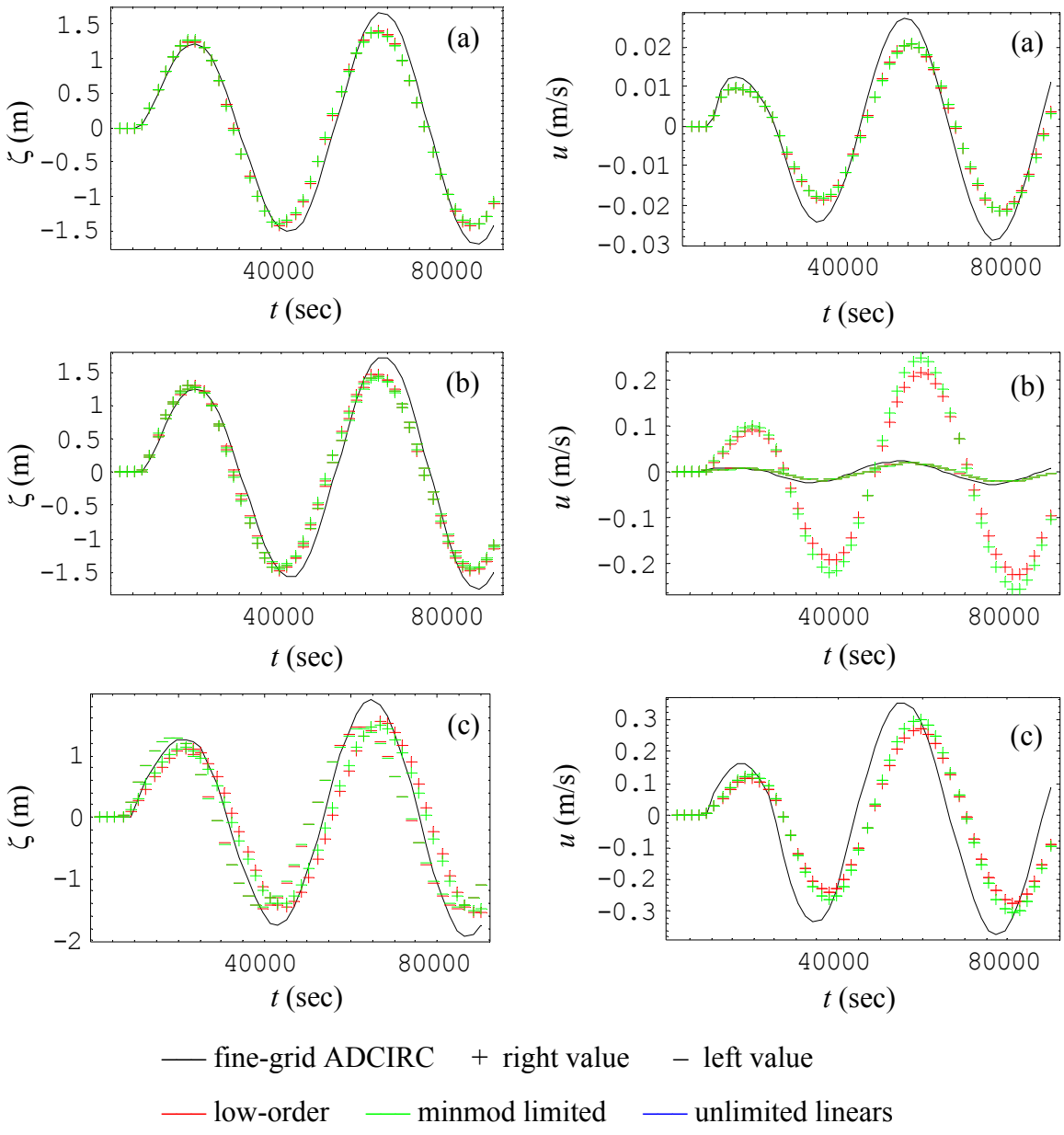


Figure 4.35 Time series at bathymetry stations for DG/CG <–>ADCIRC model with the subdomain interface located at the middle of the slope; simulated on a grid ratio of 60: a) toe of slope, b) middle of slope, c) top of slope.

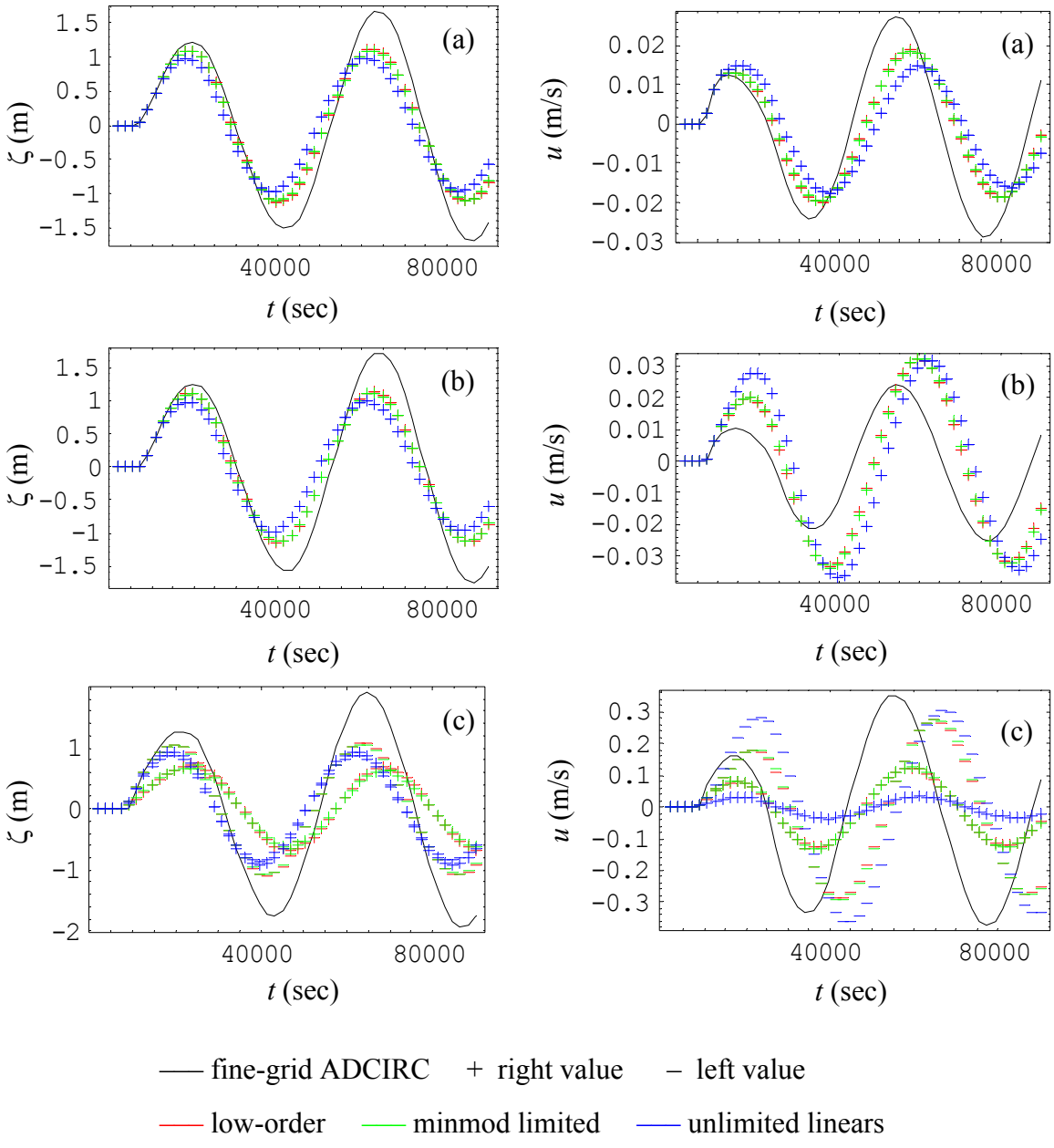


Figure 4.36 Time series at bathymetry stations for DG/CG <–> ADCIRC model with the subdomain interface located at the top of the slope; simulated on a grid ratio of 60: a) toe of slope, b) middle of slope, c) top of slope.

series at the top is discontinuous. The unlimited version is only stable at this interface location. However, none of the approximations match the fine-grid solution at any of the stations and the solutions get worse as time progresses.

Global elevation and velocity results for all three interface locations are presented in Figure 4.37. The vertical dashed lines indicate the location of the bathymetry features – toe, middle and top of slope – for the grid ratio of 60, as given in Table 4.6. Recall that the unlimited version is unstable when the interface is located at the toe or middle of the slope; and the corresponding results are missing from the global figures. As seen in the station time series, this coupling scheme is not capable of capturing the fine-grid solution at this resolution.

The elevation and velocity errors were computed for each time series by using the most accurate state value (left or right) for all comparisons that involve discontinuous data. For both state variables, the average and maximum error over the entire domain is computed for the last temporal output and summarized in Table 4.11. Additionally, the maximum and L_2 average errors over the entire time series were calculated and are summarized for all of the spatial approximations and interface locations in Table 4.12. In each table, the minimum errors for each station or over the entire domain are indicated by bold-italics in each column. Since the unlimited version is unstable for two of the interface locations, some of the corresponding table entries will be missing. Overall, this model does not perform well at this resolution.

Summary of $60\Delta x$ grid ratio results

In general, one notes that since the domain-wide ADCIRC and DG models have problems at this coarse resolution, the coupled models cannot be expected to perform well either. In the future, it would be interesting to examine other types of grids, such as the LTEA (local truncation error analysis), which would place more nodes near the subdomain interfaces and other areas where errors are concentrated. The minimum global errors from

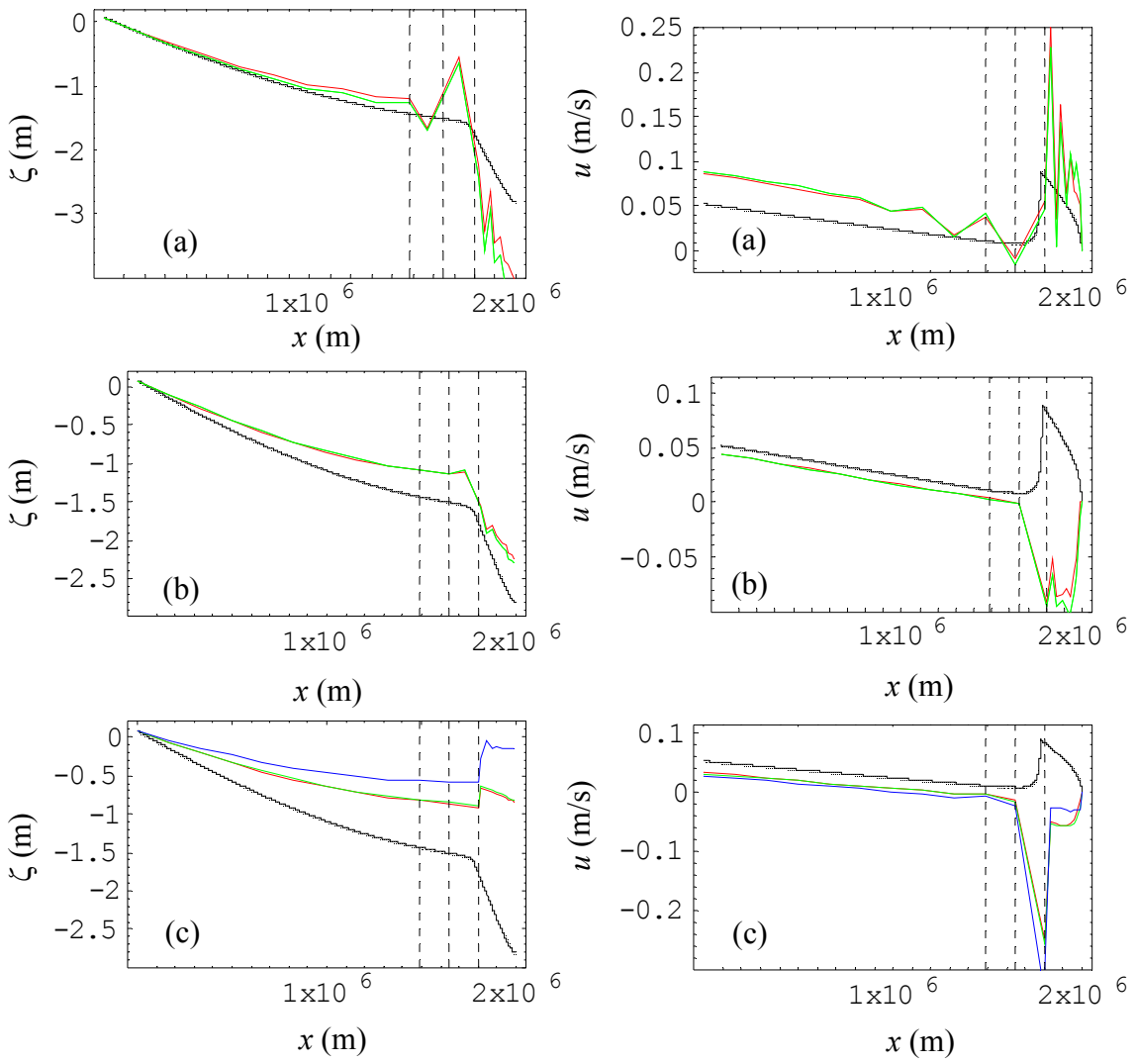


Figure 4.37 Final output at t = 90,000 seconds for DG/CG <–> ADCIRC model simulated on a grid ratio of 60 with the subdomain interface located at the: (a) toe, (b) middle, and (c) top.

Table 4.12 Station errors for DG/CG <–> ADCIRC model on a grid ratio of 60.

Spatial approximation	Interface location	Station location	L_2 errors		L_∞ errors	
			elevation	velocity	elevation	velocity
piecewise constants	toe	toe	0.2181 ^{R}	0.0315 ^{R}	0.4617 ^{R}	0.0596 ^{R}
		mid	***0.2077*** ^{L}	0.1428	0.4420 ^{L}	0.2665
		top	0.3143 ^{R}	0.1197	0.6611 ^{R}	0.2120
unlimited linears	toe	all	unstable	unstable	unstable	unstable
minmod linears	toe	toe	0.2247 ^{R}	0.0321 ^{R}	0.4831 R	0.0626 ^{R}
		mid	0.2186 ^{L}	0.1615	0.4604 ^{L}	0.3011
		top	***0.2836*** ^{R}	0.1487	0.6926 ^{R}	0.2600
piecewise constants	mid	toe	***0.1987***	***0.0048***	***0.3411***	***0.0087***
		mid	0.2135 ^{L}	***0.0056*** ^{L}	***0.3660*** ^{L}	***0.0094*** ^{L}
		top	0.3648 ^{R}	0.1108	0.6875 ^{R}	0.1881
unlimited linears	mid	all	unstable	unstable	unstable	unstable
minmod linears	mid	toe	0.2109	0.0050	0.3551	0.0092
		mid	0.2260 ^{L}	0.0060 ^{L}	0.3816 ^{L}	0.0099 ^{L}
		top	0.3118 ^{R}	***0.1067***	***0.5865*** ^{R}	***0.1791***
piecewise constants	top	toe	0.3400	0.0082	0.6318	0.0155
		mid	0.3581	0.0139	0.6660	0.0234
		top	0.4823 ^{L}	0.1543 ^{R}	0.8833 ^{L}	0.2776 ^{R}
unlimited linears	top	toe	0.4938	0.0120	0.8765	0.0211
		mid	0.5189	0.0198	0.9213	0.0331
		top	0.6324 ^{R}	0.2021 ^{R}	1.1206 ^{R}	0.3478 ^{R}
minmod linears	top	toe	0.3508	0.0085	0.6490	0.0160
		mid	0.3694	0.0143	0.6840	0.0240
		top	0.4956 ^{L}	0.1530 ^{R}	0.9055^{L}	0.2764 ^{R}

The superscripts indicate which discontinuous state value was used at the interface to compute the errors: ^{L} left or ^{R} right.
The minimum errors are denoted by bold-italics.

Table 4.11 Global errors for DG/CG <–> ADCIRC model simulated on a grid ratio of 60.

Spatial approximation	Interface location	L_2 errors		L_∞ errors	
		elevation	velocity	elevation	velocity
piecewise constants	toe	0.6980	0.0530	1.2266	0.1764
	mid	0.3585	0.0826	0.5856	0.1749
	top	1.1598	0.0961	1.9746	0.3349
unlimited linears	toe	unstable	unstable	unstable	unstable
	mid	unstable	unstable	unstable	unstable
	top	1.5927	0.1043	2.6658	0.4154
minmod linears	toe	0.8611	***0.0525***	1.5320	***0.1520***
	mid	***0.3299***	0.0903	***0.5120***	0.1791
	top	1.1791	0.0983	1.9907	0.3402

This table summarizes the global errors for the last output at t = 90,000 seconds.
The minimum errors for each column are indicated by the bold-italics type.

each of the coupled models are summarized below in Table 4.13. Overall, coupled model 2 has the lowest errors of the three coupled models and has slightly higher errors than the domain-wide DG model. In the next section, similar comparisons will be made for all of the coupled models using a finer grid before conclusions are drawn about the ideal combination of spatial approximations and interface locations.

Table 4.13 Comparison of global errors for all coupled models and a grid ratio of 60.

Coupled model	Spatial approximation	Interface location	L_2 errors		L_∞ errors	
			elevation	velocity	elevation	velocity
domain-wide DG	unlimited	na	0.0659	0.0167	0.1028	0.0282
1	constants	na	—	0.0261	—	0.0883
	minmod	na	0.1094	—	0.2236	—
2	unlimited	toe	***0.0752***	***0.0214***	***0.1152***	***0.0379***
3	constants	toe	—	0.0525	—	0.1520
	minmod	mid	0.3299	—	0.5120	—

This table summarizes the minimum global errors for all three of the coupled models for the last temporal output at t = 90,000 seconds.
More than one row may be used for a particular model when the minimum errors were not uniform over all spatial approximations or interface locations.

4.8.2 Simulation results for grid ratio of 300

ADCIRC model

For each of the three bathymetry features, the time series for the domain-wide ADCIRC model at the $L/\Delta x = 300$ resolution are shown in Figure 4.38. Notice that the solution obtained with the coarser resolution is able to capture all of the elevation and velocity behavior, with the exception of the velocity at the top of the slope. At that station, the peaks from the coarser simulation are somewhat damped relative to the fine-grid simulation. Given these ADCIRC results, it is expected that the coupled models should be able to capture most of the fine-grid behavior using this grid resolution. The global results for the ADCIRC model at this coarser grid are presented with those of the domain-wide DG model below in Figure 4.40.

Domain-wide DG model

For each of the three bathymetry features, the time series for the domain-wide DG model are shown in Figure 4.39. The plus and minus symbols indicate the discontinuous results, where the plus symbol indicates the value to the right of the element boundary and the minus symbol indicates the value to the left.

For the station at the toe of the slope, notice that all three approximations capture the elevation and velocity behavior quite well. Similarly, for the station at the middle of the slope, all three approximations capture the behavior of both the elevation and velocity. However, the right state value for the low-order and minmod versions overshoot the velocity peaks somewhat. Finally, for the station located at the top of the slope, the right state value of all three approximations is able to capture both the elevation and velocity behavior.

Figure 4.40 compares the model results over the entire domain for the final output at $t = 90{,}000$ seconds. The vertical dashed lines indicate the location of the bathymetry features – toe, middle and top of slope – for the grid ratio of 300, as given in Table 4.6. The

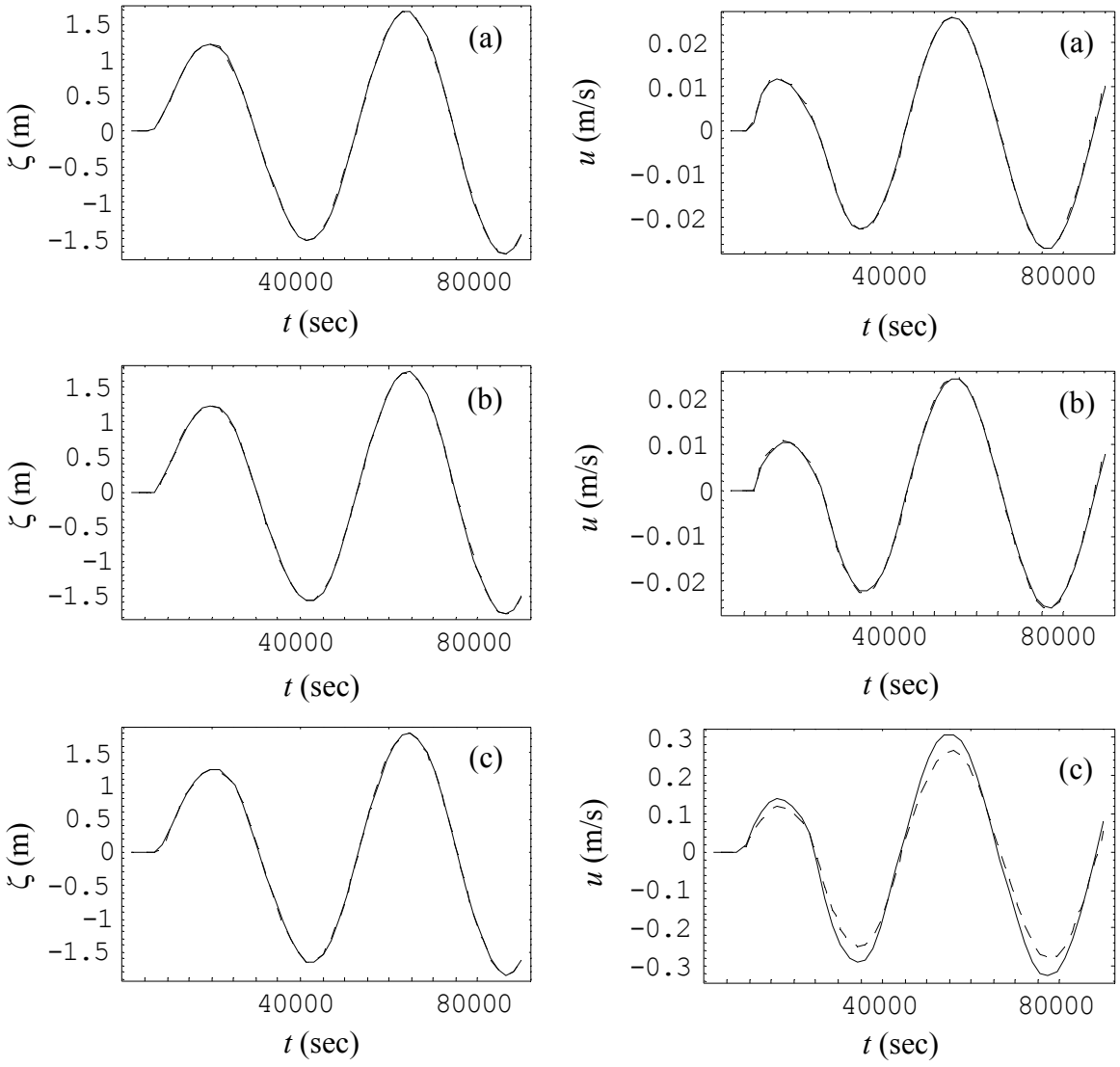


Figure 4.38 Time series at bathymetry stations for ADCIRC model simulated with a grid ratio of 300: a) toe of slope, b) middle of slope, c) top of slope.

higher-order approximations capture the global behavior better than the lower-order

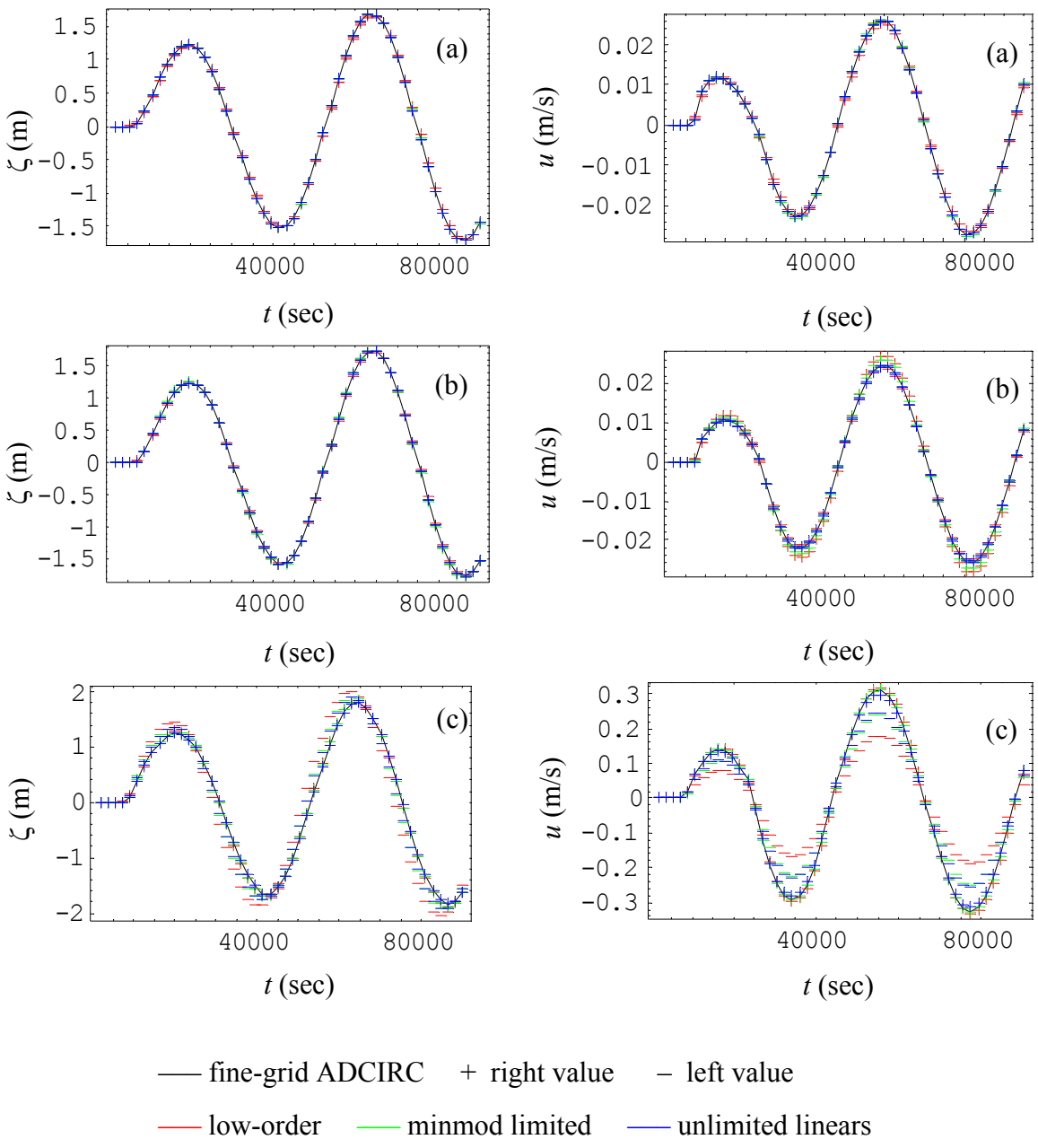


Figure 4.39 Time series at bathymetry stations for domain-wide DG model simulated with a grid ratio of 300: a) toe of slope, b) middle of slope, c) top of slope.

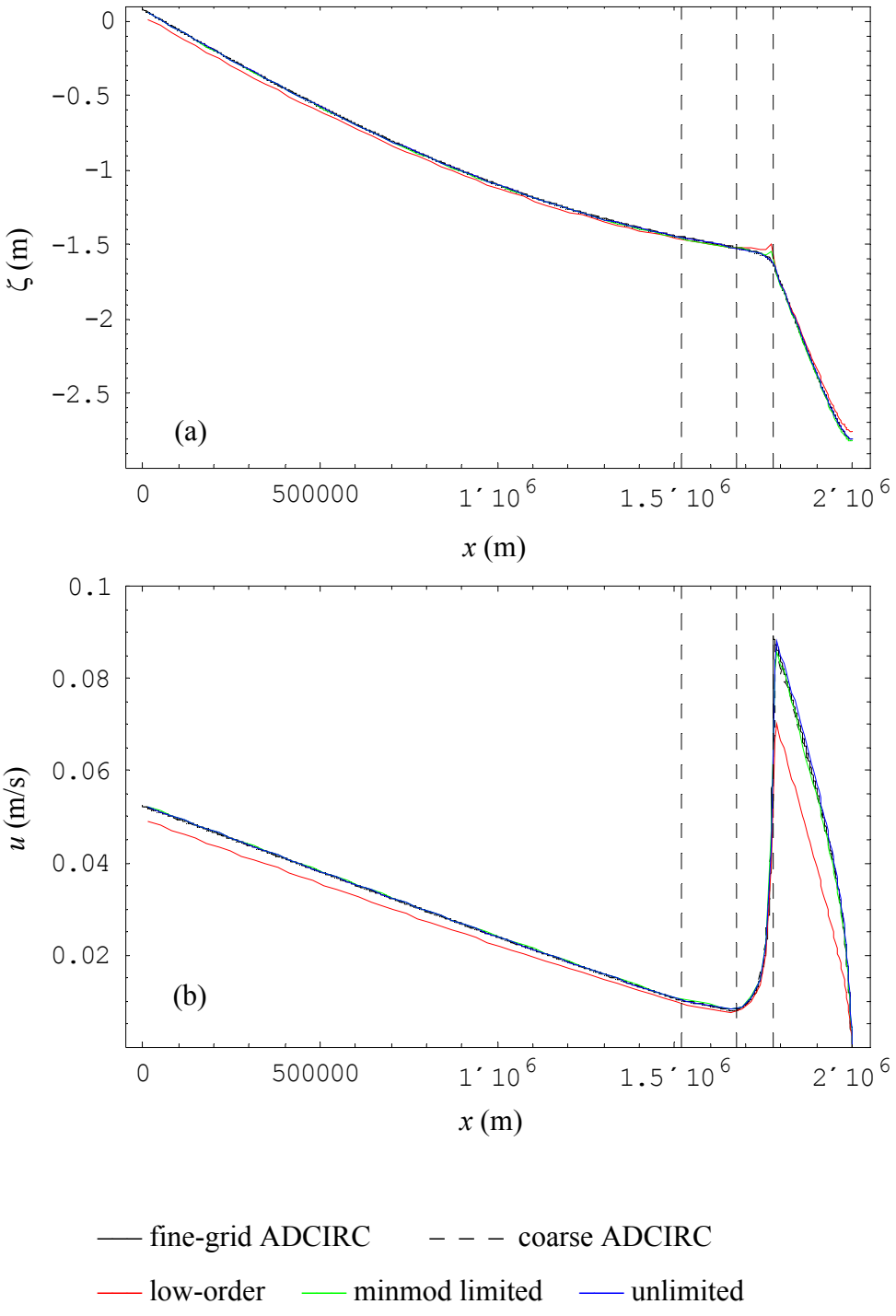


Figure 4.40 Final output at $t = 90{,}000$ seconds for domain-wide DG and ADCIRC models simulated on a grid ratio of 300: (a) elevations and (b) velocities.

version, with the unlimited version being the best.

The elevation and velocity errors were computed for each time series by using the most accurate state value (left or right) for all comparisons that involve discontinuous data. For both state variables, elevation and velocity, the maximum and L_2 average errors over the entire time series were calculated and are summarized for all of the versions in Table 4.15. Additionally, the average and maximum errors over the entire domain are computed for the last temporal output and included in the table, denoted by the word "global" in the station column. The minimum errors for each station and over the entire

Table 4.14 Spatial errors for domain-wide models simulated on a grid ratio of 300.

Spatial approximation	Station location	L_2 errors		L_∞ errors	
		elevation	velocity	elevation	velocity
domain-wide ADCIRC model					
continuous linears	toe	***0.0012***	0.0001	0.0046	0.0002
	mid	***0.0013***	0.0002	0.0041	0.0004
	top	***0.0027***	0.0298	***0.0060***	0.0505
	global	***0.0010***	0.0024	0.0050	0.0234
domain-wide DG model					
piecewise constants	toe	0.0323 ^{R}	0.0003 ^{L}	0.0570 ^{R}	0.0008 ^{L}
	mid	0.0093 ^{R}	0.0006 ^{L}	0.0280 ^{R}	0.0011 ^{L}
	top	0.0349 ^{R}	0.0090 ^{R}	0.0650 ^{R}	0.0170 ^{R}
	global	0.0345	0.0105	0.1097	0.0216
unlimited linears	toe	0.0016 ^{R}	***0.0000*** ^{L}	***0.0031*** ^{R}	***0.0001*** ^{L}
	mid	0.0019 ^{L}	***0.0001*** ^{L}	***0.0036*** ^{L}	***0.0001*** ^{L}
	top	0.0037 ^{R}	***0.0071*** ^{R}	0.0065 ^{R}	***0.0120*** ^{R}
	global	0.0028	***0.0009***	***0.0044***	***0.0052***
minmod linears	toe	0.0058 ^{L}	0.0001 ^{R}	0.0104 ^{L}	0.0003 ^{R}
	mid	0.0054 ^{L}	0.0002 ^{L}	0.0100 ^{L}	0.0004 ^{L}
	top	0.0279 ^{R}	0.0024 ^{R}	0.0472 ^{R}	0.0069 ^{R}
	global	0.0053	0.0020	0.0226	0.0078

The superscripts indicate which discontinuous state value was used to compute the errors for the domain-wide DG model: ^{L} left or ^{R} right.
The minimum errors are indicated by the bold-italics typeface.

domain are indicated by bold-italics typeface in each column.

Notice that the unlimited higher-order version minimizes the errors both locally and globally for the various domain-wide DG versions. Meanwhile, the ADCIRC model has slightly lower errors for the elevation results, although the unlimited higher-order domain-wide DG performs better for the velocity profile. Thus, the domain-wide DG model does a slightly better job, than the coarse ADCIRC model, of capturing the correct velocity behavior for the highly advective region on the shelf.

Coupled model 1

The time series for coupled model 1, at each of the three bathymetry features, are shown in Figure 4.41. The plus and minus symbols indicate the results from the coupled model, where for discontinuous results the plus symbol indicates the value to the right of the element boundary and the minus symbol indicates the value to the left. The elevations are discontinuous with this coupled model while the velocity results are continuous. The higher-order version without limiting becomes unstable as time progresses, and the results for this version are not shown.

For the stations at the toe and middle of the station, notice that both of the approximations are able to capture the elevation and velocity behavior. However, the higher-order minmod version does a slightly better job of capturing the peaks. For the station at the top of the slope, both approximations are able to capture the elevation behavior; but neither is able to capture the peaks in the velocity time series.

Figure 4.42 compares the model results over the entire domain for the final output at t = 90,000 seconds. The vertical dashed lines indicate the location of the bathymetry features – toe, middle and top of slope – for the grid ratio of 300, as given in Table 4.6. Since the unlimited linear version is unstable, no results are shown for this version. Both of the approximations are able to capture the global elevation results, although the minmod version is more accurate. Meanwhile, for the velocity, both versions are able to capture the

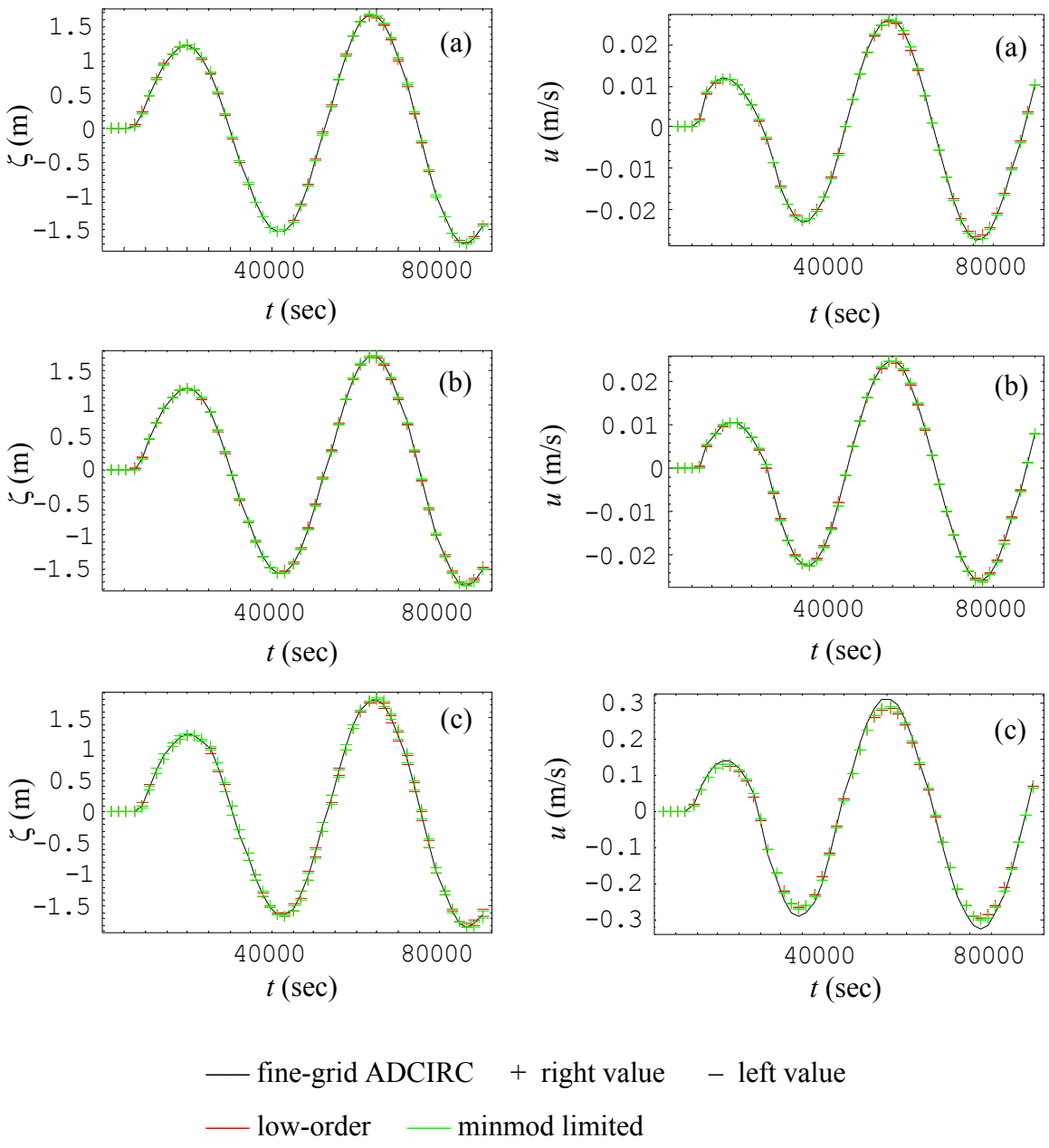


Figure 4.41 Time series at bathymetry stations for DG/CG model simulated with a grid ratio of 300: a) toe of slope, b) middle of slope, c) top of slope.

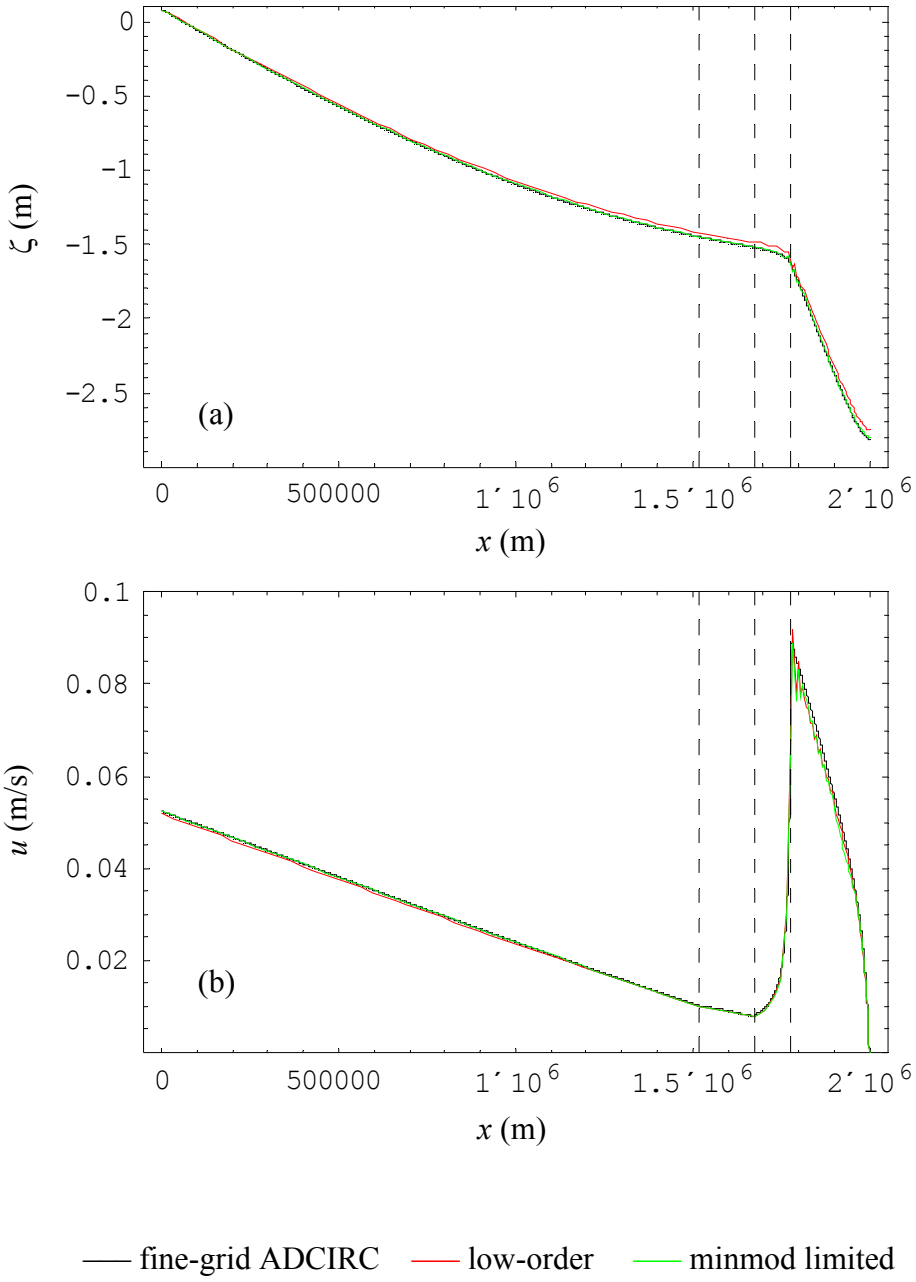


Figure 4.42 Final output at $t = 90{,}000$ seconds for DG/CG model simulated on a grid ratio of 300: (a) elevations and (b) velocities.

reflected wave, but both also introduce slight oscillations at the shelf break. This is likely due to the continuous velocity field being unable to handle the sharp front.

The elevation and velocity errors were computed for each time series by using the most accurate state value (left or right) for all comparisons that involve discontinuous data. For both state variables, the maximum and L_2 average errors over the entire time series were calculated and are summarized for all of the versions in Table 4.15. Additionally, the average and maximum errors over the entire domain are computed for the last temporal output and included in the table, denoted by the word "global" in the station column. The minimum errors for each station and over the entire domain are indicated by bold-italics.

Table 4.15 Spatial errors for DG/CG model simulated on a grid ratio of 300.

Spatial approximation	Station location	L_2 errors		L_∞ errors	
		elevation	velocity	elevation	velocity
piecewise constants	toe	0.0131 ^{R}	0.0004	0.0246 ^{R}	0.0008
	mid	0.0177 ^{R}	0.0002	0.0327 ^{R}	0.0006
	top	0.0395 ^{R}	0.0169	0.0665 ^{R}	0.0294
	global	0.0403	***0.0018***	0.0640	***0.0100***
unlimited linears	all	unstable	unstable	unstable	unstable
minmod linears	toe	***0.0022*** ^{R}	***0.0001***	***0.0052*** ^{R}	***0.0003***
	mid	***0.0043*** ^{L}	***0.0002***	***0.0090*** ^{L}	***0.0004***
	top	***0.0375*** ^{L}	***0.0150***	***0.0637*** ^{L}	***0.0254***
	global	***0.0077***	0.0022	***0.0382***	0.0122

The superscripts indicate which discontinuous state value was used to compute the errors for the coupled model: ^{L} left or ^{R} right.
The minimum errors are indicated by the bold-italics typeface.

Globally, the low-order version does slightly better for the velocities, while the minmod version does significantly better for the elevations. Additionally, the minmod version performs best for all of the stations, and appears to be the best choice for this model.

Coupled model 2

As for the previous model, the plus and minus symbols indicate the results from the coupled model, where for discontinuous results the plus symbol indicates the state value to the right of the element boundary and the minus symbol indicates the value to the left. Both the elevations and velocities are discontinuous in the DG subdomain, while both variables in the ADCIRC subdomain are continuous. Three figures will be presented for this model, one for each location of the subdomain interface.

The time series for coupled model 2 with the subdomain interface located at the toe of the slope are shown in Figure 4.43. Since the bathymetry features are to the right of the subdomain interface, all of the time series are located in the DG subdomain and are discontinuous.

For the station at the toe of the slope, notice that all three approximations are able to capture the elevation and velocity behavior. However, the right states of the higher-order versions overshoot the velocity peaks. Meanwhile, for the station at the middle of the slope, all three approximations are able to capture the elevation behavior; but only the left state of the low-order and minmod approximations are really able to capture the velocity behavior. Finally, for the station at the top of the slope, the right state values for all three approximations are able to capture the elevation behavior; but the left state of the low-order version has a slight shift and overshoots the peaks. The right states of all three approximations do a reasonable job of capturing the velocity behavior at this station, although they do overshoot the peaks slightly.

The time series for coupled model 2 with the subdomain interface located at the middle of the slope are shown in Figure 4.44. The time series at the toe is continuous since it lies in the ADCIRC subdomain and the other two time series are discontinuous.

For the station at the toe of the slope, notice that all three approximations are able to capture the elevation and velocity behavior; although the higher-order versions do a

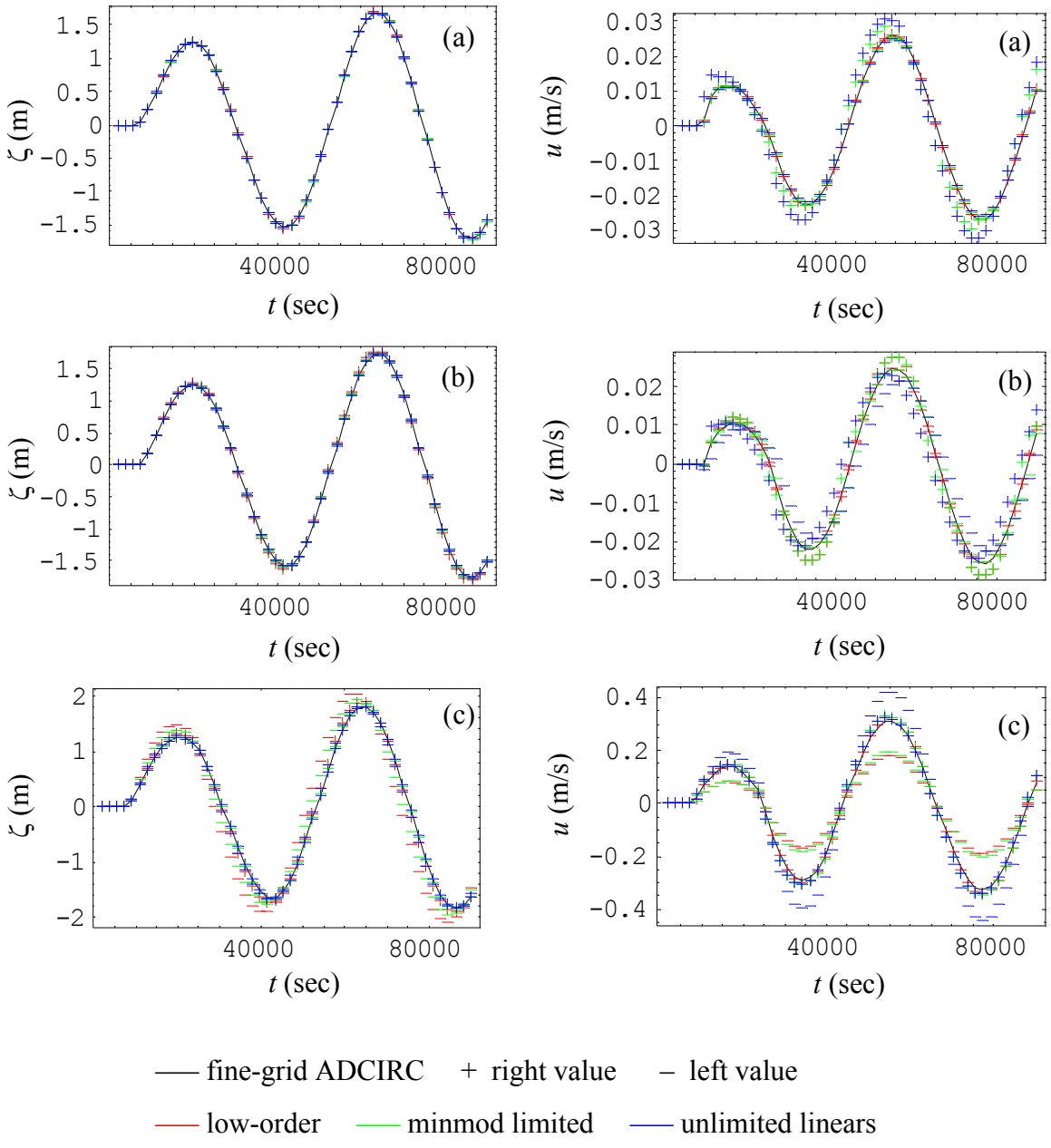


Figure 4.43 Time series at bathymetry stations for DG <–> ADCIRC model with the subdomain interface located at the toe of the slope and simulated on a grid ratio of 300: a) toe of slope, b) middle of slope, c) top of slope.

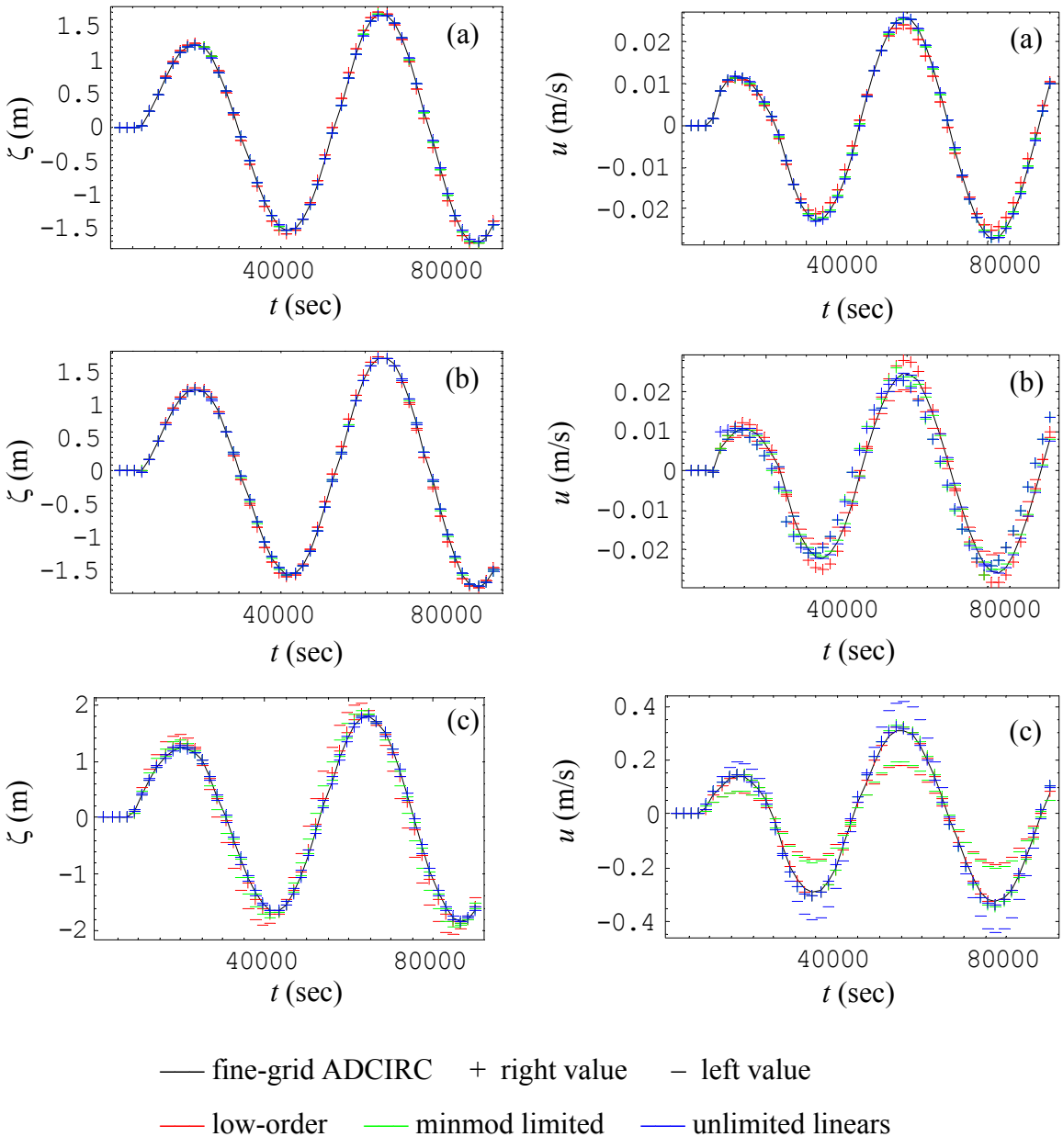


Figure 4.44 Time series at bathymetry stations for DG <–> ADCIRC model with the subdomain interface located at the middle of the slope; simulated on a grid ratio of 300: a) toe of slope, b) middle of slope, c) top of slope.

slightly better job of matching at the peaks. For the station at the middle of the slope, all three approximations are able to capture the elevation behavior; but the low-order version overshoots the peaks slightly. Meanwhile, for the velocity at this station, the left state values for the higher-order versions are able to capture the correct behavior, with the minmod version doing the best job of matching at the peaks. Finally, for the station at the top of the slope, all three approximations are able to capture the elevation behavior; but the left state values for the low-order version are shifted relative to the fine-grid solution. The right state values for all three approximations are also able to capture the velocity behavior, although the peaks are magnified slightly.

The time series for coupled model 2 with the subdomain interface located at the top of the slope are shown in Figure 4.45. The time series for the toe and middle of the slope are continuous since they are located in the ADCIRC subdomain and the time series at the top is discontinuous. None of the approximations are able to capture the elevation or velocity behavior at the toe and middle stations. The elevation series generally have a phase lag and the peaks are slightly damped, while the velocity series generally have a phase lag and the peaks are amplified. The right state values for all three approximations do a reasonable job of capturing the elevation and velocity behavior at the top of the slope.

Global elevation and velocity results for all three interface locations are presented in Figure 4.46. The vertical dashed lines indicate the location of the bathymetry features – toe, middle and top of slope – for the grid ratio of 300, as given in Table 4.6. As was seen from the station time series presented above, the global results are best when the subdomain interface is located near the toe or middle of the slope. When the interface is too near to the highly advective behavior, the coupled model is not able to capture the correct velocity behavior on the shelf, and the velocity errors propagate into the entire domain.

The elevation and velocity errors were computed for each time series by using the most accurate state value (left or right) for all comparisons that involve discontinuous data.

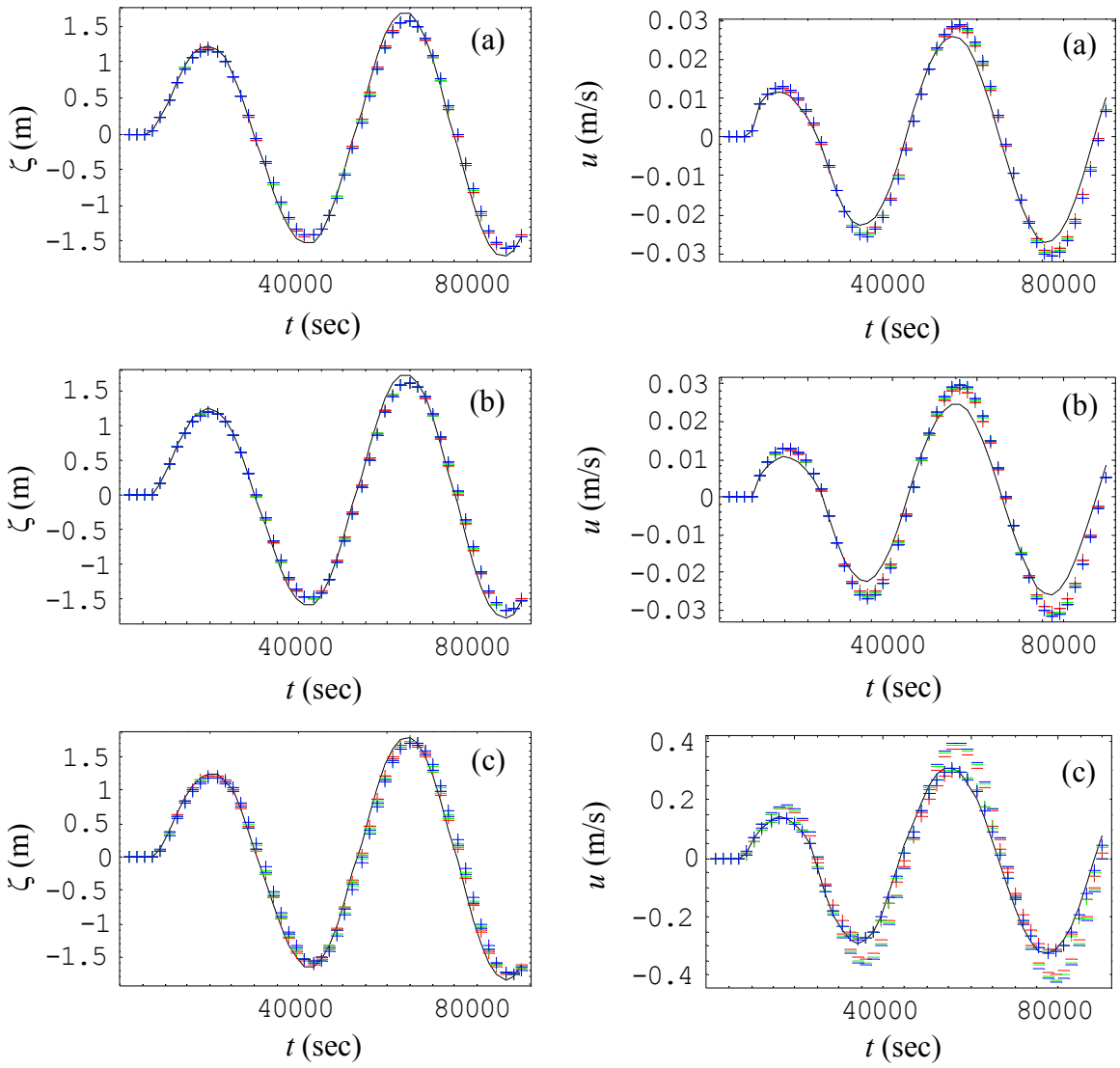


Figure 4.45 Time series at bathymetry stations for DG <–> ADCIRC model with the subdomain interface located at the top of the slope and simulated on a grid ratio of 300: a) toe of slope, b) middle of slope, c) top of slope.

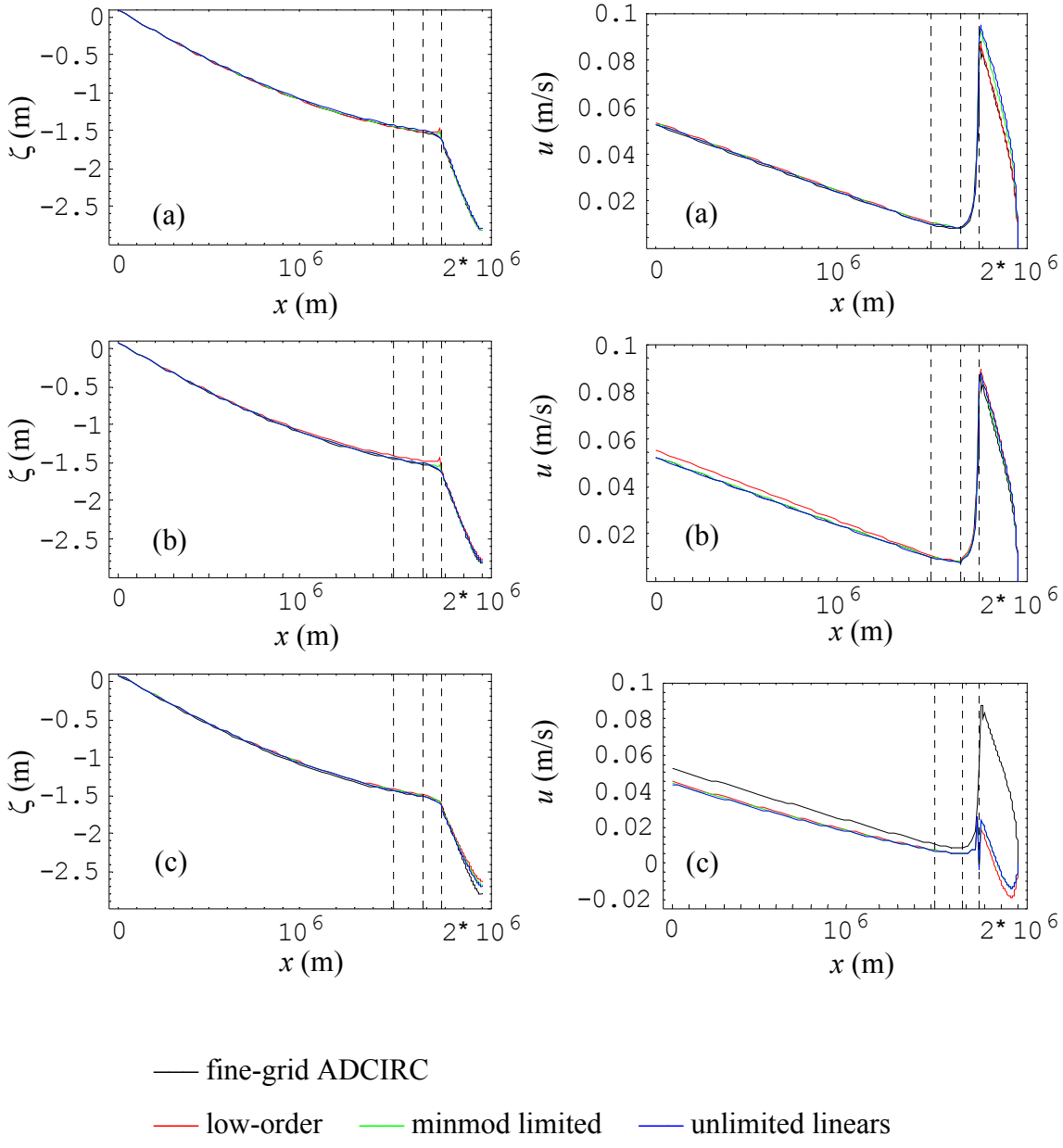


Figure 4.46 Final output at t = 90,000 seconds for DG <–> ADCIRC model simulated on a grid ratio of 300 with the subdomain interface located at the: (a) toe, (b) middle, and (c) top.

For both state variables, the maximum and L_2 average errors over the entire time series were calculated and are summarized for all of the spatial approximations and interface locations in Table 4.16. Additionally, the average and maximum errors over the entire domain are summarized for the last temporal output in Table 4.17.In each table, the minimum errors for each station or over the entire domain are indicated by the bold-italics typeface for each column.

Table 4.17 Global errors for DG <–> ADCIRC model simulated on a grid ratio of 300.

Spatial approximation	Interface location	L_2 errors		L_∞ errors	
		elevation	velocity	elevation	velocity
piecewise constants	toe	0.0161	***0.0011***	0.1414	0.0061
	mid	0.0208	0.0052	0.0299	0.0091
	top	***0.0120***	0.0036	0.0864	0.0090
unlimited linears	toe	0.0423	0.0025	0.1803	0.0077
	mid	0.0204	0.0015	***0.0299***	***0.0045***
	top	0.0162	0.0011	0.0830	0.0062
minmod linears	toe	0.0899	0.0411	0.1856	0.0785
	mid	0.0554	0.0380	0.1196	0.0813
	top	0.0648	0.0382	0.1372	0.0803

This table summarizes the global errors for the last output at t = 90,000 seconds.
The minimum errors for each column are indicated by the bold-italics type.

For the station errors, note that, in general, the elevation and velocity errors are minimized when the subdomain interface is located at the middle of the slope and higher-order spatial approximations are used. However, the most accurate elevation data at the top station is when the interface is located at the toe of the slope. Globally, the maximum errors are minimized when the unlimited version with the subdomain interface located at the middle of the slope is used. The average errors are minimized when piecewise constants are used, but there is no consistent subdomain interface location suggested by these error norms.

Table 4.16 Station errors for DG <–> ADCIRC model simulated on a grid ratio of 300.

Spatial approximation	Interface location	Station location	L_2 errors		L_∞ errors	
			elevation	velocity	elevation	velocity
piecewise constants	toe	toe	0.0145 ^{L}	0.0003^{R}	0.0273 ^{L}	0.0006 ^{R}
		mid	0.0218 ^{L}	0.0006 ^{L}	0.0434 ^{L}	0.0011 ^{L}
		top	0.0805 ^{R}	0.0079 ^{R}	0.1401 ^{R}	0.0156 ^{R}
unlimited linears	toe	toe	0.0153^{R}	0.0007 ^{L}	0.0295 ^{R}	0.0013 ^{L}
		mid	0.0141 ^{R}	0.0040 ^{L}	0.0275 ^{R}	0.0068 ^{L}
		top	***0.0037*** ^{R}	0.0181 ^{R}	***0.0089*** ^{R}	0.0280 ^{R}
minmod linears	toe	toe	0.0109 ^{R}	0.0005 ^{L}	0.0211 ^{R}	0.0010 ^{L}
		mid	0.0211 ^{L}	0.0017 ^{L}	0.0402 ^{L}	0.0050 ^{L}
		top	0.0308 ^{R}	0.0173 ^{R}	0.0543 ^{R}	0.0295 ^{R}
piecewise constants	mid	toe	0.0579	0.0015	0.1056	0.0028
		mid	0.0547 ^{R}	0.0018 ^{R}	0.1007 ^{R}	0.0031 ^{R}
		top	0.0832 ^{R}	***0.0073*** ^{R}	0.1492 ^{R}	***0.0152*** ^{R}
unlimited linears	mid	toe	***0.0076***	***0.0001***	***0.0131***	***0.0002***
		mid	0.0051 ^{R}	***0.0002*** ^{L}	0.0097 ^{R}	***0.0005*** ^{L}
		top	0.0170 ^{R}	0.0150 ^{R}	0.0289 ^{R}	0.0260 ^{R}
minmod linears	mid	toe	0.0105	0.0002	0.0196	0.0004
		mid	***0.0037*** ^{R}	0.0004 ^{L}	***0.0071*** ^{R}	0.0007 ^{L}
		top	0.0175 ^{R}	0.0151 ^{R}	0.0295 ^{R}	0.0250 ^{R}
piecewise constants	top	toe	0.0903	0.0025	0.1764	0.0048
		mid	0.0965	0.0033	0.1894	0.0060
		top	0.1052 ^{R}	0.0315 ^{R}	0.2098 ^{R}	0.0603 ^{R}
unlimited linears	top	toe	0.1113	0.0031	0.2161	0.0058
		mid	0.1188	0.0040	0.2315	0.0073
		top	0.1299 ^{L}	0.0192 ^{R}	0.2534 ^{L}	0.0387 ^{R}
minmod linears	top	toe	0.1036	0.0029	0.2012	0.0054
		mid	0.1106	0.0037	0.2157	0.0068
		top	0.1209 ^{L}	0.0231 ^{R}	0.2367 ^{L}	0.0509 ^{R}

The superscripts indicate which discontinuous state value was used to compute the error for the discontinuous data: ^{L} left or ^{R} right.
The minimum errors for each station are indicated by the bold-italics type.

Coupled model 3

As for the previous models, the plus and minus symbols indicate the results from the coupled model, where the plus symbol indicates the state value to the right of the element boundary and the minus symbol indicates the value to the left. Only the elevations are discontinuous in the DG/CG subdomain. Meanwhile, both the elevation and velocity results in the ADCIRC subdomain are continuous and the velocity over the entire domain is discontinuous only at the subdomain interface. The higher-order versions, unlimited and minmod limited, are both unstable for this grid ratio and no results are presented for these versions at any of the interface locations. (Time steps as small as $\Delta t = 10^{-4}$ seconds were attempted without any success.) Therefore, only one figure will be presented for this model, which will compare the location of the subdomain interface for the low-order version.

The time series for coupled model 3 using low-order DG approximating spaces and various locations for the subdomain interface are shown in Figure 4.47. When the interface is located at the toe, the bathymetry features are to the right of the subdomain interface; and all of the elevation time series are discontinuous. Meanwhile, when the interface is located at the middle of the slope, the elevation time series at the toe is continuous since it lies in the ADCIRC subdomain and the other two elevation series are discontinuous. Finally, when the interface is located at the top of the slope, the elevation time series for the toe and middle of the slope are continuous since they are located in the ADCIRC subdomain and the elevation series at the top is discontinuous. In this figure, the different colors represent subdomain interface locations, not spatial approximations, as in all of the other figures thus far.

For the station at the toe of the slope, notice that the elevation and velocity behavior are captured when the subdomain interface is located at the toe or middle of the slope, but not when it is located at the top. Meanwhile, for the station at the middle of the slope, notice that the elevation behavior is captured when the subdomain interface is located at the toe

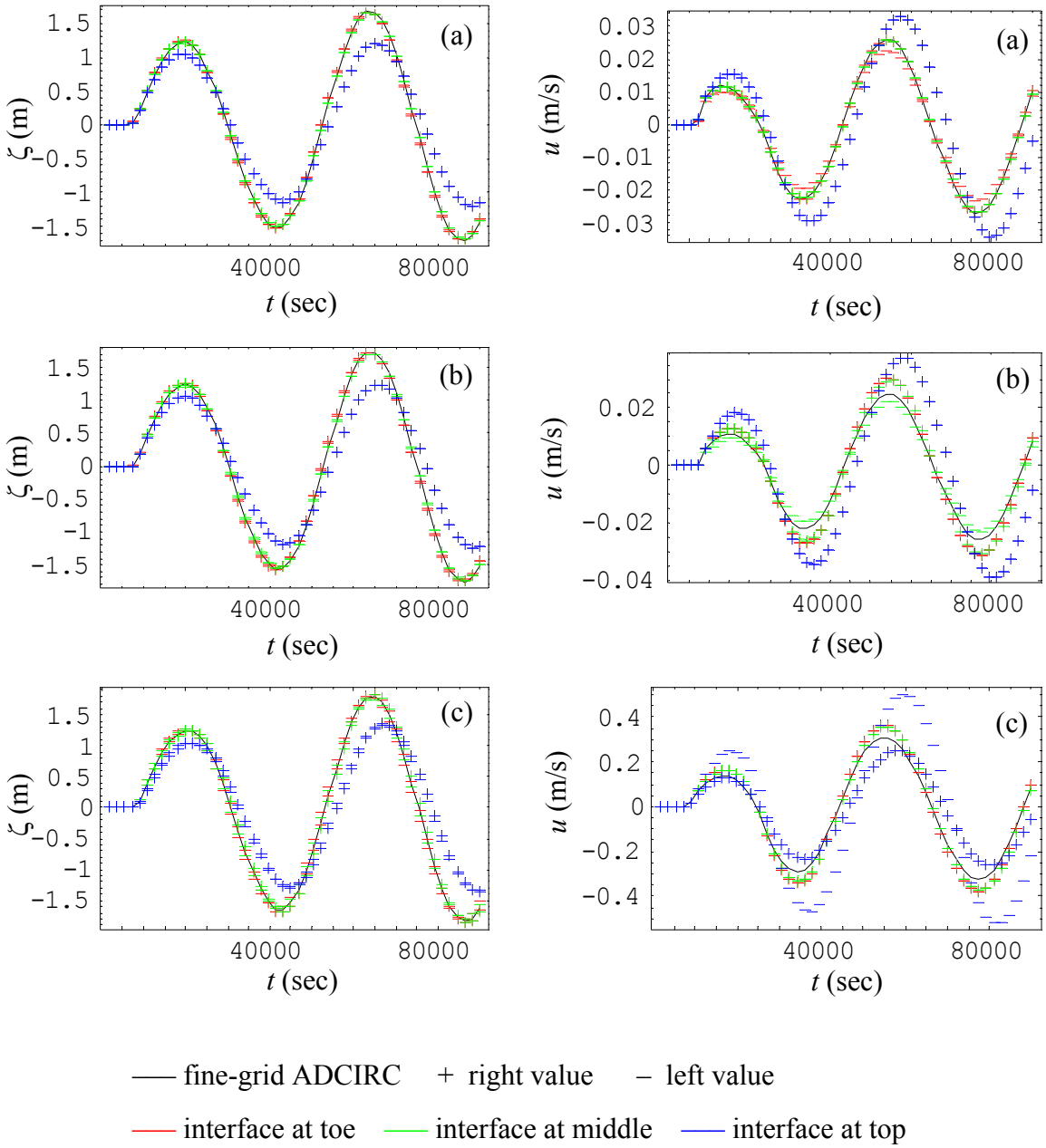


Figure 4.47 Time series at bathymetry stations for DG/CG <–> ADCIRC model using low-order spatial approximations with various subdomain interface locations and simulated on a grid ratio of 300: a) toe of slope, b) middle of slope, c) top of slope.

or middle, but not when it is located at the top. Furthermore, at this station, the velocity behavior is captured only when the subdomain interface is located at the middle of the slope. Finally, for the station at the top of the slope, the elevation behavior is captured when the subdomain interface is located at either the toe or middle of the slope, but not when it is at the top of the slope. However, even with the subdomain interface located at the toe or middle, the velocity behavior is not adequately captured – the peaks are amplified relative to the fine-grid solution.

Global elevation and velocity results for all three interface locations are presented in Figure 4.48. The vertical dashed lines indicate the location of the bathymetry features – toe, middle and top of slope – for the grid ratio of 300, as given in Table 4.6. Recall that the higher-order versions are unstable for this grid resolution, such that the corresponding results are missing from the figures. Notice that the global results with the subdomain interface located at the toe of the slope are a pretty good match to the fine-grid ADCIRC solution, while the results with the interface located at the top do not capture the correct behavior for either variable.

The elevation and velocity errors were computed for each time series by using the most accurate state value (left or right) for all comparisons that involve discontinuous data. For both state variables, the maximum and L_2 average errors over the entire time series were calculated and are summarized for the low-order spatial approximations and all three interface locations in Table 4.18. Additionally, the average and maximum errors over the entire domain are computed for the last temporal output and included in this table. The minimum errors for each station or over the entire domain are indicated by bold-italics in each column.

Graphically, it appears that this model is most accurate for the elevations when the subdomain interface is located at the middle of the slope, while the velocity solution is optimized with the subdomain interface located at the toe of the slope. Numerically, the

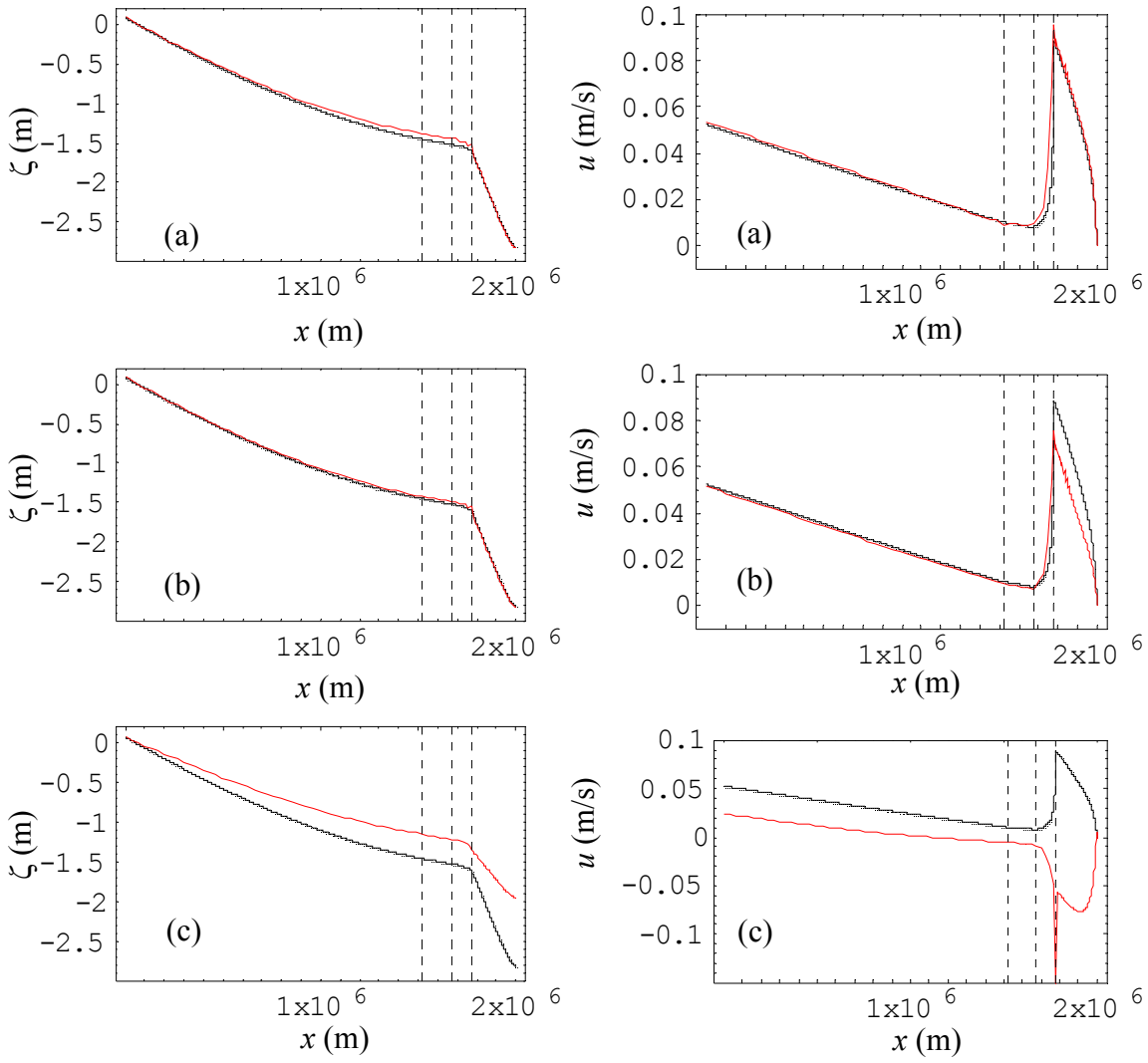


Figure 4.48 Final output at t = 90,000 seconds for DG/CG <–> ADCIRC model simulated on a grid ratio of 300 with the subdomain interface located at the: (a) toe, (b) middle, and (c) top.

Table 4.18 Spatial errors for DG/CG <–> ADCIRC model on a grid ratio of 300.

Spatial approximation	Interface location	Station location	L_2 errors		L_∞ errors	
			elevation	velocity	elevation	velocity
piecewise constants	toe	toe	0.0494 ^{R}	***0.0003*** ^{R}	0.0853 ^{R}	***0.0007*** ^{R}
		mid	0.0546 ^{L}	0.0033	0.0940 ^{L}	0.0056
		top	***0.0292*** ^{R}	0.0314	***0.0557*** ^{R}	0.0540
		global	0.0360	***0.0047***	0.0912	0.0390
piecewise constants	mid	toe	***0.0213***	0.0005	***0.0383***	0.0010
		mid	***0.0225*** ^{L}	***0.0018*** ^{L}	***0.0399*** ^{L}	***0.0030*** ^{L}
		top	0.0548 ^{L}	***0.0287***	0.0866 ^{L}	***0.0497***
		global	***0.0187***	0.0105	***0.0537***	***0.0229***
piecewise constants	top	toe	0.3378	0.0094	0.6356	0.0174
		mid	0.3620	0.0120	0.6800	0.0214
		top	0.4073 ^{L}	0.0727 ^{R}	0.7633 ^{L}	0.1343 ^{R}
		global	0.4965	0.0863	0.8710	0.2931

The superscripts indicate which state value was used to compute the errors for the discontinuous data: ^{L} left or ^{R} right.
The minimum errors for each column are indicated by the bold-italics type.

errors seem to indicate that the interface should be located at the middle of the slope in order to reduce most of the station and global errors.

Summary of 300Δx grid ratio results

The minimum global errors from each of the coupled models are summarized below in Table 4.19. The upper portion of the table repeats the results from the $L/\Delta x = 60$ grid resolution that were presented in Table 4.13 in the previous section. For each portion of the table, the overall minimum results for each column are indicated by bold-italics, while the minimum results from the coupled models alone are indicated with italics.

Notice that the errors for the domain-wide DG and ADCIRC models are significantly reduced when comparing the results for the 60Δx and 300Δx grid ratios. This improved accuracy also carries over to the coupled models. In general, coupled model 2 has the minimum global errors relative to the other coupled models. However, the domain-wide

Table 4.19 Comparison of minimum global errors for all three coupled models.

Coupled model	Spatial approximation	Interface location	L_2 errors		L_∞ errors	
			elevation	velocity	elevation	velocity
Grid ratio of 60						
ADCIRC	continuous linear	na	***0.0649***	0.0191	0.1571	0.0814
DG	unlimited	na	0.0659	***0.0167***	***0.1028***	***0.0282***
1	constants	na	—	0.0261	—	0.0883
	minmod	na	0.1094	—	0.2236	—
2	unlimited	toe	*0.0752*	*0.0214*	*0.1152*	*0.0379*
3	minmod	toe	—	0.0525	—	0.1520
		mid	0.3299	—	0.5120	—
Grid ratio of 300						
ADCIRC	continuous linear	na	***0.0010***	0.0024	0.0050	0.0234
DG	unlimited	na	0.0028	***0.0009***	***0.0044***	0.0052
1	constants	na	—	0.0018	—	0.0100
	minmod	na	*0.0077*	—	0.0382	—
2	constants	toe	—	*0.0011*	—	—
		top	0.0120	—	—	—
	unlimited	mid	—	—	*0.0299*	***0.0045***
3	constants	toe	—	0.0047	—	—
		mid	0.0187	—	0.0537	0.0229

More than one row is used for a particular model when the minimum errors were not uniform over all spatial approximations or interface locations.
The minimum errors comparing the coupled and domain-wide models are indicated in bold-italics, while the minimum errors comparing only the coupled models are indicated by italics.

models have lower errors than this coupled model, indicating that some accuracy is lost with the coupling procedure. The magnitude of these differences is very small, however, and the coupled model essentially performs as well as the domain-wide models. The following section will compare the CPU times for the coupled models and domain-wide ADCIRC and DG models.

4.8.3 CPU times for East coast simulations

Using the maximum stable time step, the CPU usage for each of the variations

(spatial approximation, grid ratio, and subdomain interface location) were computed for approximately three tidal cycles – 135,000 seconds of simulation time – and are summarized below in Table 4.20. Recall that the higher-order versions of coupled model 3 were unstable for the grid ratio of 300, therefore only the CPU usage for the low-order version is presented for this model and grid ratio.

As expected, the CPU usage is increased as the DG workload increases due to higher order interpolants and slope limiting; i.e., for most of the coupled or domain-wide DG models with a fixed grid ratio and interface location, the CPU usage increases as you move from piecewise constants to unlimited linears to minmod limited linears. For the domain-wide DG model, the workload is approximately doubled for each of these transitions when the same time step is used for all three interpolants. This trend is not realized for all of the coupled model combinations, since the unlimited version often requires a much smaller time step than the low-order and minmod limited versions; and therefore requires more CPU time than either of the other two spatial approximations. However, for all of the coupled model combinations, it is always true that the higher-order versions require more CPU time than the low-order versions.

Theoretically, one would also expect that as the subdomain interface moves towards the top of the slope, the CPU usage should be reduced since the DG subdomain contains fewer elements. However, since the maximum stable time steps also change with the interface location, this is not easily discerned from the data in the above table. Due to the complicated interplay between the subdomains, this will not always be true. For example, the maximum stable time step allowed for the DG or DG/CG subdomain in coupled models 2 and 3 may be significantly smaller than that allowed for the domain-wide ADCIRC model. This means that the ADCIRC subdomain will be performing suboptimally and that the CPU usage may actually increase as the interface moves towards the top of the slope, since more matrix computations will be required in the now larger ADCIRC subdomain.

Table 4.20 Comparison of CPU usage for east coast simulations.

Spatial approximation	Interface location	max stable Δt (sec)		CPU usage (sec)	
		grid ratio 60	grid ratio 300	grid ratio 60	grid ratio 300
ADCIRC model					
linears	na	500.0	100.0	< 0.00001	0.0625
domain-wide DG model					
constants	na	500.0	100.0	< 0.00001	0.0156
unlimited	na	20.0	5.0	0.0156	0.3906
minmod	na	20.0	5.0	0.0469	0.7813
coupled model 1: DG/CG					
constants	na	500.0	100.0	< 0.00001	0.0313
unlimited	na	1.0	—	0.4063	—
minmod	na	100.0	1.0	0.0156	3.2031
coupled model 2: DG $\rightarrow$ ADCIRC					
piecewise constants	toe	100.0	100.0	< 0.00001	0.0313
	mid	200.0	100.0	< 0.00001	0.0313
	top	200.0	100.0	0.0156	0.0313
unlimited linears	toe	50.0	10.0	0.0313	0.3906
	mid	50.0	10.0	0.0313	0.4063
	top	10.0	10.0	0.1406	0.4219
minmod linears	toe	100.0	10.0	0.0156	1.3125
	mid	500.0	10.0	< 0.00001	1.1719
	top	200.0	10.0	0.0156	1.0000
coupled model 3: DG/CG $\rightarrow$ ADCIRC					
piecewise constants	toe	500.0	100.0	< 0.00001	0.0313
	mid	500.0	100.0	< 0.00001	0.0469
	top	500.0	100.0	< 0.00001	0.0469
unlimited linears	toe	—	—	—	—
	mid	—	—	—	—
	top	0.001	—	1562.98	—
minmod linears	toe	10.0	—	0.1719	—
	mid	100.0	—	0.0156	—
	top	10.0	—	0.1719	—
fine-grid ADCIRC – 8192 elements grid ratio 60 – 20 elements grid ratio 300 – 107 elements (For the fine-grid ADCIRC solution: max stable Δt = 0.1 sec, CPU usage = 501.625 sec.)					

For coupled model 2 with a grid ratio of 300, notice that the same time step is used for all of the interface locations for a given spatial approximation. For piecewise constants, the CPU time does not change at all as the interface is moved from the toe of the slope to the top. Meanwhile, for the unlimited linear version the CPU time actually increases as the interface makes this transition. The work being shifted to the ADCIRC subdomain at the suboptimal time step (Δt max is one-tenth of that allowed for the domain-wide ADCIRC model) is increasing more than the work in the DG subdomain is reduced. Finally, for the minmod limited version, one notes that there is a slight decrease in CPU time for the interface transitions, as expected. This indicates that the minmod limiter is more costly than running ADCIRC at a suboptimal time step.

For coupled model 3 with a grid ratio of 300 and piecewise constant approximations, notice that the maximum stable time step is the same as that allowed for the domain-wide ADCIRC and low-order DG models, namely 100.0 seconds. This allows for an interesting comparison of workloads. At this time step, the ADCIRC model uses 0.0625 seconds of CPU time while the low-order domain-wide DG model uses 0.0156 seconds. Meanwhile, as might be expected, the coupled model use some value in between these. For coupled model 3 the workload goes from 0.0313 seconds to 0.0469 seconds as the interface moves. As more work is transferred to the more costly ADCIRC subdomain, the CPU times increase slightly. However, notice that the domain-wide DG model is only computationally less expensive for piecewise constants. For higher-order spatial approximations, the time step is much more limited and the workload increases significantly.

In general, notice that due to the reduction in time step required for the coupled models to remain stable, the CPU usage is increased relative to the domain-wide DG model. This defeats the purpose of using the coupled models, namely to reduce computational costs, while maintaining accuracy. Although the CPU times for the best coupled scheme, coupled model 2, are only slightly higher than those for the domain-wide DG. Further

studies using full non-linear models with flux coupling, and perhaps having different time steps in each subdomain, would be required to see if there is combination of workloads that would minimize the CPU times while maintaining accuracy. Most importantly, the DG workloads will increase *significantly* in 2D and 3D applications as the number of edges, which is where the extra computations from the discontinuities arise, grows faster than the number of elements.

4.9 Conclusions and recommendations

Each of the coupled models was ranked for seven specific characteristics: 1) Propagation behavior weighted at 10%, 2) spatial accuracy weighted at 10%, 3) temporal accuracy weighted at 10%, 4) flat bottom simulations weighted at 10%, 5) variable bathymetry simulations weighted at 20%, 6) CPU usage weighted at 20% and 7) stability weighted at 20%. The last three characteristics exhibit the robustness of the model in actual applications and were thus weighted more heavily. The rankings go from 1 to 9 with 1 being the worst and 9 being the best. The rank for each characteristic is multiplied by its weighting factor and summed over all 7 characteristics in order to reach a final score for each model (maximum possible score = 9.0). The model with the highest score is deemed to have the "best" behavior over all of the characteristics. Table 4.21 summarizes the ranking and final score of each coupled model.

From this table, coupled model 2 using piecewise constants has the best score overall, followed closely by the unlimited higher-order version of this model. Coupled model 1 using piecewise constants has the next best score, followed by the minmod version of coupled model 2. The remaining models have significantly lower scores, with the unlimited version of coupled model 3 being the worst.

Although the coupled models using the linear system of SWE and explicit overlapping subdomain coupling are simplified, several conclusions can be drawn from this study: 1) the subdomain coupling methods results in higher spatial and temporal errors,

Table 4.21 Qualitative summary of coupled model behavior.

Spatial approximation	Propagation behavior	Spatial error	Temporal error	Flat bottom simulations	East coast simulations	CPU times	Stability	Final score
weighting factors	10%	10%	10%	10%	20%	20%	20%	(max = 9)
Coupled model 1 — DG/CG								
constants	1	7	9	7	6	8	6	6.4
unlimited	8	8	3	8	3	2	3	4.3
minmod	9	9	4	9	5	4	4	5.7
Coupled model 2 — DG → ADCIRC								
constants	5	4	8	5	7	9	9	7.2
unlimited	6	6	5	6	9	6	8	6.9
minmod	7	5	6	4	8	5	7	6.2
Coupled model 3 — DG/CG → ADCIRC								
constants	2	3	7	1	4	7	5	4.5
unlimited	3	1	1	2	1	1	1	1.3
minmod	4	2	2	3	2	3	2	2.5

which is most likely due to the explicit overlapping coupling at the interface; 2) the subdomain interface should be located below the shelf break to minimize errors, namely at or below the middle of the slope; 3) for equation coupling, as used in the DG/CG and $\mathrm{DG/CG} \rightarrow \mathrm{ADCIRC}$ models, use of interface averaging to interpolate to the CG grid space results in an unstable algorithm, while L_2 interpolation remains stable; and 4) the higher-order versions of the equation coupled schemes (DG/CG and $\mathrm{DG/CG} \rightarrow \mathrm{ADCIRC}$) have severe stability restrictions, and in some cases it was not possible to obtain a stable solution with them.

From these observations, and due to its promising behavior with the simplified explicit overlapping element coupling, it is recommended that coupled model 2, $\mathrm{DG} \rightarrow \mathrm{ADCIRC}$, be examined further in a full non-linear scheme with flux coupling at the subdomain interface. Additionally, a second-order Runge-Kutta temporal discretization should be examined for the higher-order linear DG approximations.

CHAPTER 5

Local Discontinuous Galerkin Finite Element Discretization of the Transport Equation in Quasi-Three-Dimensional Domains

5.1 Introduction

Transport of a dissolved species in groundwater or surface water is governed by the advection-dispersion equation. A mass conservative discretization of this equation is essential for accurate prediction of contaminant transport in water quality applications, and for salinity and temperature transport in prognostic baroclinic hydrodynamic applications. Such problems are often advection-dominated and solution techniques must be able to model sharp fronts while retaining stability and mass conservation.

This study seeks to explore the interaction of discontinuous discretizations for the advection-dispersion equation within the continuous hydrodynamic model, ADCIRC (ADvanced 3D CIRCulation model *Luettich et al.* [1992, 2003]). Traditionally, the ADCIRC model uses a mode splitting technique with a continuous finite element discretization to solve for the elevations and velocities independently rather than solving a completely coupled system. In this framework, the 2D depth-averaged GWC equation is solved to obtain the new elevations, which are then fed into the 3D momentum equation to obtain the new horizontal velocities at all depths. Finally, these horizontal results are used in the 3D continuity equation to obtain the new vertical velocities. For testing purposes, a

2D *x-z* ADCIRC model is implemented in this study (for development details see *Dresback and Kolar* [2004a]). The same mode splitting technique is utilized, but now the 1D GWC equation is solved to obtain the changes in elevation and then the 2D *x-z* momentum equation is used to determine the horizontal velocities. Finally, these results are utilized within an *x-z* continuity equation to obtain the vertical velocities. For more details about the ADCIRC hydrodynamic model see the referenced reports.

Hench and Luettich [1997] found that continuous Galerkin (CG) discretizations of the transport equation coupled with the two-dimensional depth-integrated (2DDI) ADCIRC hydrodynamic model did not conserve species mass and that the time scale was severely restricted in order to maintain stability. In contrast, discontinuous methods are known to be conservative in their formulation and are capable of handling sharp fronts. Discontinuous Galerkin (DG) algorithms differ from traditional CG finite element methods in that the primitive equations are integrated over individual elements, rather than the entire global domain, and the state values are allowed to be discontinuous at element interfaces. This framework allows different orders of approximating spaces to be used in adjacent elements (local *p* refinement) without the use of mortar spaces since each element is integrated independently of its neighbors; thus the algorithm can easily adapt and incorporate higher resolution (i.e., higher-order basis functions) near shocks in the solution. With these favorable properties in mind, namely conservation and shock capturing without oscillations, DG methods look promising for incorporating a transport algorithm within the ADCIRC hydrodynamic model.

All DG schemes for first-order equations are equivalent; however, special techniques must be used to handle second-order diffusive terms within DG algorithms, and several schemes have been developed over the years. The technique that we will use herein is a modification of the local discontinuous Galerkin (LDG) method, as developed by *Cockburn and Shu* [1998b]. The main idea of the LDG method is to rewrite the second-order equation into a system of first-order equations and then discretize the resulting system

using the Runge-Kutta Discontinuous Galerkin (RKDG) method. The RKDG discretization for non-linear hyperbolic problems was developed and analyzed in a series of papers (*Cockburn and Shu* [1989, 1991, 1998a], Cockburn et al. [1989, 1990]) and gets its name because it uses a Runge-Kutta time-stepping scheme, with built-in projection limiting, for the temporal discretization. *Aizinger et al.* [2001] applied the LDG approach to contaminant transport in porous medium for a variety of 1D and 2D test problems, including two-component contaminant transport and non-linear adsorption. They showed that the method performs well for smooth solutions as well as sharp fronts and remains stable for higher degree polynomials.

Other discontinuous methods for diffusive problems include various penalty formulations and mixed finite element methods. *Siegel et al.* [1997], solved the advection-dispersion equation using a combination of discontinuous and mixed finite elements. They discretize the advection terms using discontinuous finite elements with slope limiters and use approximate Riemann solvers for the flux discontinuities and then use a mixed hybrid finite element method for the diffusive terms. With their method they are able to preserve sharp fronts, while avoiding excessive numerical diffusion and oscillations. They test their method over a range of Peclet numbers (1 to 10000) to verify its ability to handle diffusive and advective flows.

Oden et al. [1998] developed a discontinuous Galerkin method for diffusion operators, in which the solution and its derivatives are discontinuous across element boundaries. Their scheme is element wise conservative and supports *h*-, *p*- and *hp*-refinement. They impose weak continuity requirements on the interelement interfaces instead of using auxiliary variables. Although their formulation resembles mixed and interior penalty methods, no penalty terms or Lagrange multipliers are required.

Rivière et al. [1999] analyze the DG method for diffusion, as presented by *Oden et al.* [1998], as well as a nonsymmetric interior penalty Galerkin (NIPG) formulation and a

nonsymmetric constrained Galerkin (NCG) formulation. The NIPG formulation is a modification of earlier work, such that the penalty term is only required to be positive rather than bounded below by a problem-dependent constant. The NCG formulation requires that the jumps on each edge of the triangulation must have an integral average of zero. They derive *a priori hp* error estimates for each formulation. In a second paper [2001], they examine higher dimensions and present some numerical convergence results in addition to the *a priori* estimates.

Dawson and Proft [2001] applied interior penalty versions of the LDG method to transport equations. They examined two penalty formulations, one which penalizes only the jumps in the primary variable and another that penalizes the jumps in both the primary variable and the flux. When only the jumps in the primary variable are penalized, they show that one can obtain an estimate of order $h^{k+1/2}$ for a penalty parameter of order one and an estimate of order h^{k+1} for a penalty parameter of order h^{-1}, where k is the minimal degree polynomial used in the approximating functions.

Dawson and Proft [2002a] also examined coupled continuous and discontinuous methods for transport problems. In that work they apply the LDG method in one region and use a standard CG method in the other region, with suitable transmission conditions specified at the interface. They applied the resulting coupled scheme to 1D test cases with smooth solutions and sharp fronts. In each case they found that the coupled algorithm transmits the solution through the coupling interface and performs better than a pure CG method.

5.2 Transport model in an x-z domain

In order to study three-dimensional transport processes without large scale computational efforts, this study uses a quasi 3D domain, namely a laterally averaged slice in *x-z* coordinates where the *x* coordinate is the horizontal direction and the *z* coordinate is along the depth of the water column. The resulting 2D *x-z* transport equation in

conservative form without reactions, sources or sinks is given as

$$\frac{\partial C}{\partial t} + \frac{\partial}{\partial x}\left[uC - N_x\frac{\partial C}{\partial x}\right] + \frac{\partial}{\partial z}\left[wC - N_z\frac{\partial C}{\partial z}\right] = 0 \qquad \textbf{(5.1)}$$

where $C(x, z, t)$ is the concentration of solute in moles per unit volume in the fluid phase (or for salinity, grams of salt per kilogram of seawater denoted as $^0/_{00}$), $u(x, z, t)$ and $w(x, z, t)$ are the x and z components of the fluid velocity vector, and N_x and N_z are the x and z diffusion coefficients, which are assumed to be constant for this study. As the ultimate goal is to couple the DG discretization of the transport equation with the existing ADCIRC hydrodynamic model, which uses CG approximations, the velocity fields for the transport algorithm (u and w) will be approximated with continuous functions. Equation (5.1) is solved on a domain Ω for $t > 0$ with the initial condition

$$C(x, z, 0) = C^0(x, z), \qquad x, z \in \Omega\ . \qquad \textbf{(5.2)}$$

Three different types of boundary conditions are discussed below in §5.3.3.

Following the framework of the 3D ADCIRC hydrodynamic model, where the horizontal and vertical discretizations are treated separately, this study assumes that the horizontal and vertical dependencies within the transport equation are also separable. Thus, the concentration and velocity vectors can be written as

$$C(x, z, t) = C(t)\ \varphi(x)\ \psi(z), \qquad \textbf{(5.3)}$$

$$u(x, z, t) = u(t)\ \phi(x)\ \zeta(z), \text{ and} \qquad \textbf{(5.4)}$$

$$w(x, z, t) = w(t)\ \phi(x)\ \zeta(z) \qquad \textbf{(5.5)}$$

where $\varphi(x)$ and $\psi(z)$ are allowed to be discontinuous approximating spaces and $\phi(x)$ and $\zeta(z)$ are continuous spaces. In keeping with the derivation of the ADCIRC model, whose velocities will be used within the transport model, linear Lagrange functions are used for

the velocity bases in the transport model. The choice of discontinuous functions will be treated in §5.3 and options for boundary conditions will be treated in §5.3.3. In summary, the complete spatial discretization consists of a 1D integration in the *x* direction ignoring the *z* dependencies of the variables and then another 1D integration in the *z* direction on the resulting semi-discrete equations. Finally, a forward Euler temporal discretization is applied.

5.3 LDG discretization of x-z transport

The discontinuous discretization technique that is examined in this study is a modified LDG scheme. The spatial discretization follows the work of *Cockburn and Shu* [1998b], but as a first step, a simple forward Euler scheme is used instead of the original Runge-Kutta at the temporal discretization. Working from the above separable assumption, we first discretize Equation (5.1) in the *x* direction.

5.3.1 Integration in the x direction

In preparation for the *x* integrations, the second-order equation is written as a system of first-order equations in *x*:

$$\frac{\partial C}{\partial t} + \frac{\partial}{\partial x}[uC + \Upsilon] + \frac{\partial}{\partial z}\left[wC - N_z\frac{\partial C}{\partial z}\right] = 0 \qquad \textbf{(5.6)}$$

$$\tilde{\Upsilon} + \frac{\partial C}{\partial x} = 0 \qquad \textbf{(5.7)}$$

$$\Upsilon = N_x\tilde{\Upsilon} \qquad \textbf{(5.8)}$$

where Υ and $\tilde{\Upsilon}$ are auxiliary variables.

Before proceeding with the derivation, some notation is introduced. On any spatial domain R let $(\ ,\ )_R$ denote the $L^2(R)$ inner product and $\langle\ ,\ \rangle_R$ denote integration over

boundaries. Let $\{T_h\}_{h>0}$ denote a family of finite element partitions of the domain such that no element Ω_e crosses the domain boundaries, where h is the maximal element diameter. Let

$$W_{h,e} = \{\varphi : \varphi \text{ is a polynomial of degree} \le k_e \text{ on each element } \Omega_e \text{ in } T_h\}. \quad \textbf{(5.9)}$$

Note that the degree k_e can vary from one element to the next. Let n_e denote the unit outward normal to $\partial\Omega_e$. Then, for $x \in \partial\Omega_e$ define

$$\varphi^-(x) = \lim_{s \to 0^-} \varphi(x + sn_e) \quad \textbf{(5.10)}$$

and

$$\varphi^+(x) = \lim_{s \to 0^+} \varphi(x + sn_e). \quad \textbf{(5.11)}$$

Thus, φ^- is the value of φ to the "left" of the boundary, assuming the normal n_e points from "left" to "right". Then approximations c, γ, $\tilde{\gamma} \in W_{h,e}$ are sought for C, Υ, $\tilde{\Upsilon}$ on each element Ω_e.

Next each equation is multiplied by a test function and integrated over an element to get, after integrating by parts,

$$(\frac{\partial c}{\partial t},\varphi)_{\Omega_e} - (uc + \gamma,\frac{d\varphi}{dx})_{\Omega_e} + \langle (uc + \gamma) \cdot n_e, \varphi^- \rangle_{\partial\Omega_e} + (\frac{\partial}{\partial z}\left[wc - N_z\frac{\partial c}{\partial z}\right],\varphi)_{\Omega_e} = 0 \quad \textbf{(5.12)}$$

$$(\tilde{\gamma},\varphi)_{\Omega_e} - (c,\frac{d\varphi}{dx})_{\Omega_e} + \langle c \cdot n_e, \varphi^- \rangle_{\partial\Omega_e} = 0 \quad \textbf{(5.13)}$$

$$(\gamma,\varphi)_{\Omega_e} = (N_x\tilde{\gamma},\varphi)_{\Omega_e}. \quad \textbf{(5.14)}$$

An element Ω_e is defined from $x_{i-1/2}$ to $x_{i+1/2}$, where $\Delta x_i = x_{i+1/2} - x_{i-1/2}$. Note that in general the test functions, φ, in Equations (5.12)-(5.14) can be different for each equation, but for

simplicity equal order spaces are used herein for the entire system of first-order equations.

Since the approximations are allowed to be discontinuous at the boundaries, the boundary integrals must be defined. Following *Aizinger et al.* [2001], upwinding is used for c in the advection term and averaging is used for c and γ in the other boundary integrals. The upwind value is defined as

$$c^u = \begin{cases} c^- & \text{if } \mathrm{u} \cdot n_e \geq 0 \\ c^+ & \text{if } \mathrm{u} \cdot n_e < 0 \end{cases} \tag{5.15}$$

and the average value is defined as

$$c^a = \frac{1}{2}(c^- + c^+) \tag{5.16}$$

where c^- and c^+ are as defined by Equations (5.10) and (5.11). Also note that since the same approximating spaces are used in Equations (5.13) and (5.14), Equation (5.14) can be substituted directly into (5.13) to eliminate one of the equations before solving the system. With these changes, the semi-discrete equations in the x direction given by the LDG method are defined to be the solution of the following weak formulation:

$$(\frac{\partial c}{\partial t},\varphi)_{\Omega_e} - (uc + \gamma,\frac{d\varphi}{dx})_{\Omega_e} + \langle (uc^u + \gamma^a) \cdot n_e, \varphi^- \rangle_{\partial\Omega_e} + (\frac{\partial}{\partial z}\left[wc - N_z\frac{\partial c}{\partial z}\right],\varphi)_{\Omega_e} = 0 \tag{5.17}$$

$$\frac{1}{N_x}(\gamma,\varphi)_{\Omega_e} - (c,\frac{d\varphi}{dx})_{\Omega_e} + \langle c^a \cdot n_e, \varphi^- \rangle_{\partial\Omega_e} = 0\,. \tag{5.18}$$

For each element integration, the integration limits are transformed to dimensionless ξ coordinates and the basis functions are defined in these coordinates. At this point, hat notation is introduced to indicate that the coefficients are still functions of z and t and the separable approximations for c, γ, u and w are substituted into Equations (5.17)

and (5.18). The discontinuous approximations are defined to be polynomials of degree *K,* while the continuous approximations are linear Lagrange, as defined in the hydrodynamics. The approximations are given as

$$c(x, z, t) = \sum_{l=0}^{K} \hat{c}_{i,k}^{(l,m)} \varphi_l(\xi)$$

$$\gamma(x, z, t) = \sum_{l=0}^{K} \hat{\gamma}_{i,k}^{(l,m)} \varphi_l(\xi)$$

$$u(x, z, t) = \hat{u}_{i-1/2,k} \phi_0(\xi) + \hat{u}_{i+1/2,k} \phi_1(\xi)$$

$$w(x, z, t) = \hat{w}_{i-1/2,k} \phi_0(\xi) + \hat{w}_{i+1/2,k} \phi_1(\xi)$$

where the φ_l are discontinuous Legendre polynomials whose coefficients are evaluated at the center of element Ω_e in the discrete grid and the m degrees of freedom will be defined during the z integration. Index notation for the continuous velocity fields and discontinuous transport species on an x-z element are shown in Figure 5.1, and notation for linear approximating functions in the transformed coordinate space is shown in Figure 5.2.

After making these approximations for the velocities and transport solute, the weak form for the x direction approximation is given as

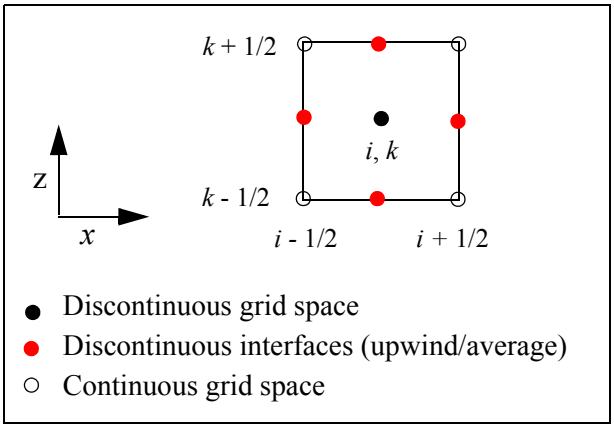


Figure 5.1 Grid notation for element (*i, k*).

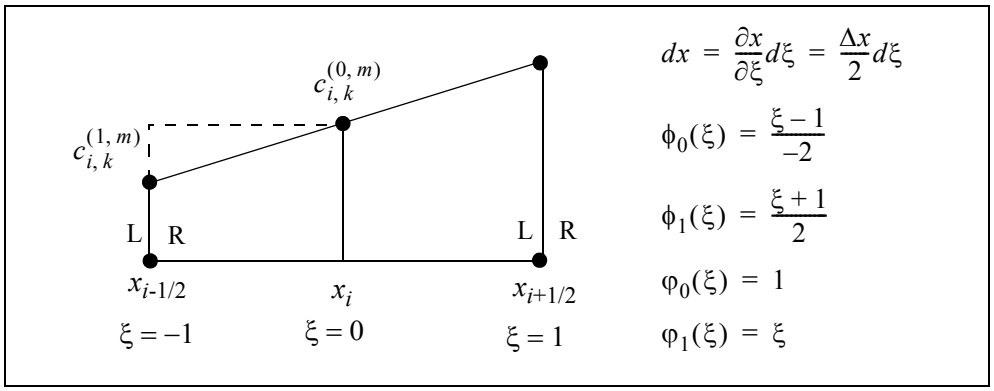


Figure 5.2 Transformed coordinate space in x direction with linear basis functions for element (i, k).

$$\sum_{l=0}^{K} \left\{ \frac{d}{dt}(\hat{c}_{i,k}^{(l,m)} \varphi_l, \varphi_j)_{\Omega_e} - \left(\hat{\gamma}_{i,k}^{(l,m)} \varphi_l, \frac{d\varphi_j}{dx}\right)_{\Omega_e} - \right.$$

$$((\hat{u}_{i-1/2,k}\,\phi_0 + \hat{u}_{i+1/2,k}\,\phi_1)\hat{c}_{i,k}^{(l,m)} \varphi_l, \frac{d\varphi_j}{dx})_{\Omega_e} +$$

$$(\hat{u}_{i+1/2,k}\ \hat{c}_{i+1/2,k}^{(u,m)} + \hat{\gamma}_{i+1/2,k}^{(a,m)})\varphi_j^- - (\hat{u}_{i-1/2,k}\ \hat{c}_{i-1/2,k}^{(u,m)} + \hat{\gamma}_{i-1/2,k}^{(a,m)})\varphi_j^- +$$

$$\left. (\frac{\partial}{\partial z}[(\hat{w}_{i-1/2,k}\,\phi_0 + \hat{w}_{i+1/2,k}\,\phi_1)\hat{c}_{i,k}^{(l,m)} \varphi_l], \varphi_j)_{\Omega_e} + (\frac{\partial}{\partial z}\left[-N_z \frac{\partial}{\partial z}\hat{c}_{i,k}^{(l,m)} \varphi_l\right], \varphi_j)_{\Omega_e} = 0 \right\} \quad \textbf{(5.19)}$$

$$\sum_{l=0}^{K} \left\{ \frac{1}{N_x}(\hat{\gamma}_{i,k}^{(l,m)} \varphi_l, \varphi_j)_{\Omega_e} - (\hat{c}_{i,k}^{(l,m)} \varphi_l, \frac{d\varphi_j}{dx})_{\Omega_e} + \hat{c}_{i+1/2,k}^{(a,m)} \varphi_j^- - \hat{c}_{i-1/2,k}^{(a,m)} \varphi_j^- = 0 \right\} \quad \textbf{(5.20)}$$

where for brevity the explicit functional dependence of the bases is not shown. After integration, the resulting system of equations for each degree of freedom $j = 0,...,K$,where K is the order of the approximation, is the semi-discrete approximation in the x direction. In this study, only linear test functions (K=1) have been examined, which gives

i = 0

$$\Delta x_i \frac{d}{dt}\hat{c}_{i,k}^{(0,m)} + (\hat{u}_{i+1/2,k}\hat{c}_{i+1/2,k}^{(u,m)} + \hat{\gamma}_{i+1/2,k}^{(a,m)}) - (\hat{u}_{i-1/2,k}\hat{c}_{i-1/2,k}^{(u,m)} + \hat{\gamma}_{i-1/2,k}^{(a,m)}) +$$

$$\frac{\Delta x_i}{2}\frac{\partial}{\partial z}\left[\hat{w}_{i-1/2,k}\left(\hat{c}_{i,k}^{(0,m)} - \frac{1}{3}\hat{c}_{i,k}^{(1,m)}\right) + \hat{w}_{i+1/2,k}\left(\hat{c}_{i,k}^{(0,m)} + \frac{1}{3}\hat{c}_{i,k}^{(1,m)}\right)\right] -$$

$$\frac{\Delta x_i}{2}\frac{\partial}{\partial z}\left[N_z\frac{\partial}{\partial z}(2\hat{c}_{i,k}^{(0,m)})\right] = 0 \qquad \textbf{(5.21)}$$

$$\frac{\Delta x_i}{N_x}\hat{\gamma}_{i,k}^{(0,m)} + \hat{c}_{i+1/2,k}^{(a,m)} - \hat{c}_{i-1/2,k}^{(a,m)} = 0 \qquad \textbf{(5.22)}$$

i = 1

$$\frac{\Delta x_i}{3}\frac{d}{dt}\hat{c}_{i,k}^{(1,m)} - \left[\hat{u}_{i-1/2,k}\left(\hat{c}_{i,k}^{(0,m)} - \frac{1}{3}\hat{c}_{i,k}^{(1,m)}\right) + \hat{u}_{i+1/2,k}\left(\hat{c}_{i,k}^{(0,m)} + \frac{1}{3}\hat{c}_{i,k}^{(1,m)}\right) + 2\hat{\gamma}_{i,k}^{(0,m)}\right] +$$

$$(\hat{u}_{i+1/2,k}\hat{c}_{i+1/2,k}^{(u,m)} + \hat{\gamma}_{i+1/2,k}^{(a,m)}) + (\hat{u}_{i-1/2,k}\hat{c}_{i-1/2,k}^{(u,m)} + \hat{\gamma}_{i-1/2,k}^{(a,m)}) +$$

$$\frac{\Delta x_i}{2}\frac{\partial}{\partial z}\left[\frac{1}{3}\hat{w}_{i-1/2,k}(-\hat{c}_{i,k}^{(0,m)} + \hat{c}_{i,k}^{(1,m)}) + \frac{1}{3}\hat{w}_{i+1/2,k}(\hat{c}_{i,k}^{(0,m)} + \hat{c}_{i,k}^{(1,m)})\right] -$$

$$\frac{\Delta x_i}{2}\frac{\partial}{\partial z}\left[N_z\frac{\partial}{\partial z}\left(\frac{2}{3}\hat{c}_{i,k}^{(1,m)}\right)\right] = 0 \qquad \textbf{(5.23)}$$

$$\frac{\Delta x_i}{3N_x}\hat{\gamma}_{i,k}^{(1,m)} - 2\hat{c}_{i,k}^{(0,m)} + \hat{c}_{i+1/2,k}^{(a,m)} + \hat{c}_{i-1/2,k}^{(a,m)} = 0 \qquad \textbf{(5.24)}$$

The first entry in the superscripts denote the degrees of freedom in the *x* direction, or the unknowns, for the discontinuous solution and can take on values of 0 or 1, indicating the degree of freedom (DOF), and *a* or *u*, indicating that either the average or upwind value should be used at the element interface discontinuity. The *m* degrees of freedom will be determined in the following discretization for the *z* direction.

5.3.2 Integration in the z direction

To proceed with the z direction discretization, Equations (5.21) and (5.23) are now written as a system of first-order equations in z, as was done in the x direction. This results in the introduction of two new equations for each j

$$\tilde{\chi}_{i,k}^{(j,m)} + \frac{\partial \hat{c}_{i,k}^{(j,m)}}{\partial z} = 0 \text{ and} \tag{5.25}$$

$$\chi_{i,k}^{(j,m)} = N_z \tilde{\chi}_{i,k}^{(j,m)} \tag{5.26}$$

where again, since subsequently the same test functions will be used for the entire system, Equation (5.26) can be substituted directly into Equation (5.25) and there is only one additional equation for each j, given as

$$\frac{1}{N_z}\chi_{i,k}^{(j,m)} + \frac{\partial \hat{c}_{i,k}^{(j,m)}}{\partial z} = 0. \tag{5.27}$$

To complete the transformation to a system of equations in z, the following replacements must be made in Equations (5.21) and (5.23) respectively:

$$-N_z \frac{\partial}{\partial z}(2\hat{c}_{i,k}^{(0,m)}) = 2\chi_{i,k}^{(0,m)} \text{ and}$$

$$-N_z \frac{\partial}{\partial z}\left(\frac{2}{3}\hat{c}_{i,k}^{(1,m)}\right) = \frac{2}{3}\chi_{i,k}^{(1,m)}.$$

Next, each equation is multiplied by the z weight function and integrated over an element. An element Ω_e is now defined from $z_{k-1/2}$ to $z_{k+1/2}$, with $\Delta z_k = z_{k+1/2} - z_{k-1/2}$ and again the integration limits are transformed to dimensionless ξ coordinates.

At this point, the hat notation is removed to indicate that the coefficients are functions of t only. Again the discontinuous bases are defined to be polynomials of degree K and the continuous bases are linear Lagrange functions such that the approximations are

given as

$$\hat{c}_{i,k}^{(j,m)}(z,t) = \sum_{l=0}^{K} c_{i,k}^{(j,l)} \psi_l(\xi)$$

$$\hat{\gamma}_{i,k}^{(j,m)}(z,t) = \sum_{l=0}^{K} \gamma_{i,k}^{(j,l)} \psi_l(\xi)$$

$$\chi_{i,k}^{(j,m)}(z,t) = \sum_{l=0}^{K} \chi_{i,k}^{(j,l)} \psi_l(\xi)$$

$$u(z,t) = u_{i,k-1/2}\,\zeta_0(\xi) + u_{i,k+1/2}\,\zeta_1(\xi)$$

$$w(z,t) = w_{i,k-1/2}\,\zeta_0(\xi) + w_{i,k+1/2}\,\zeta_1(\xi)$$

where the ψ_l are discontinuous Legendre polynomials whose coefficients are evaluated at the center of element Ω_e in the discrete grid (refer to Figure 5.1 on page 233). Notation for linear approximating functions in the transformed coordinate space is shown below in Figure 5.3.

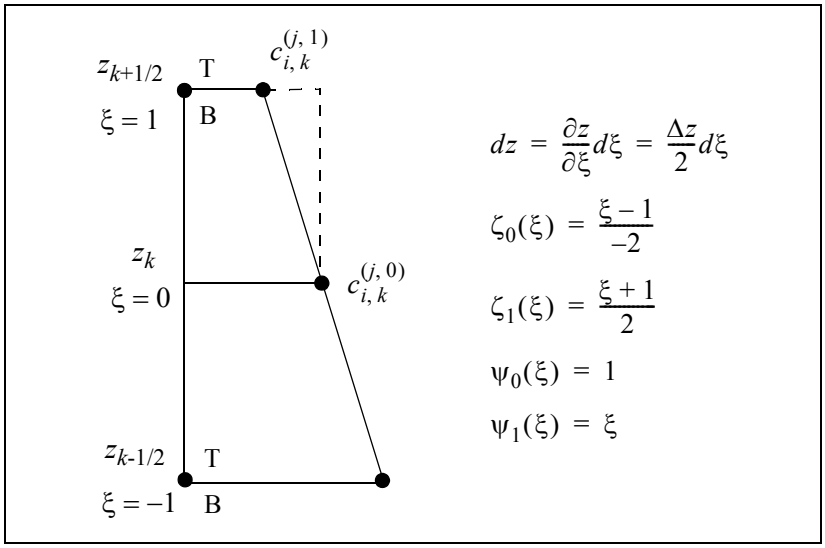


Figure 5.3 Transformed coordinate space in z direction with linear basis functions for element (i, k).

The same conventions are used for the discontinuities at the z interfaces (upwinding and averaging) as were used in the x discretization. The resulting system of weak equations in z for each j degree of freedom is given in Table 5.1, where again for brevity the explicit functional dependence of the bases are not shown.

Table 5.1 Weak equations for the z direction semi-discretization

$j = 0$
$\sum_{l=0}^{K} \left\{ \Delta x_i \frac{d}{dt}(c_{i,k}^{(0,l)} \psi_l, \psi_m)_{\Omega_e} + \right.$ $((u_{i+1/2,k-1/2}\zeta_0 + u_{i+1/2,k+1/2}\zeta_1) c_{i+1/2,k}^{(u,l)} \psi_l, \psi_m)_{\Omega_e} + (\gamma_{i+1/2,k}^{(a,l)} \psi_l, \psi_m)_{\Omega_e} -$ $((u_{i-1/2,k-1/2}\zeta_0 + u_{i-1/2,k+1/2}\zeta_1) c_{i-1/2,k}^{(u,l)} \psi_l, \psi_m)_{\Omega_e} - (\gamma_{i-1/2,k}^{(a,l)} \psi_l, \psi_m)_{\Omega_e} -$ $\frac{\Delta x_i}{2}\left((w_{i-1/2,k-1/2}\zeta_0 + w_{i-1/2,k+1/2}\zeta_1)\left(c_{i,k}^{(0,l)} - \frac{1}{3}c_{i,k}^{(1,l)}\right)\psi_l, \frac{d\psi_m}{dz}\right)_{\Omega_e} -$ $\frac{\Delta x_i}{2}\left((w_{i+1/2,k-1/2}\zeta_0 + w_{i+1/2,k+1/2}\zeta_1)\left(c_{i,k}^{(0,l)} + \frac{1}{3}c_{i,k}^{(1,l)}\right)\psi_l + 2\chi_{i,k}^{(0,l)}\psi_l, \frac{d\psi_m}{dz}\right)_{\Omega_e} +$ $\frac{\Delta x_i}{2}\left[w_{i-1/2,k+1/2}\left(c_{i,k+1/2}^{(0,u)} - \frac{1}{3}c_{i,k+1/2}^{(1,u)}\right) + w_{i+1/2,k+1/2}\left(c_{i,k+1/2}^{(0,u)} + \frac{1}{3}c_{i,k+1/2}^{(1,u)}\right)\right]\psi_l^- -$ $\frac{\Delta x_i}{2}\left[w_{i-1/2,k-1/2}\left(c_{i,k-1/2}^{(0,u)} - \frac{1}{3}c_{i,k-1/2}^{(1,u)}\right) + w_{i+1/2,k-1/2}\left(c_{i,k-1/2}^{(0,u)} + \frac{1}{3}c_{i,k-1/2}^{(1,u)}\right)\right]\psi_l^- +$ $\left. \frac{\Delta x_i}{2}(2\chi_{i,k+1/2}^{(0,a)}\psi_l^- - 2\chi_{i,k-1/2}^{(0,a)}\psi_l^-) \right\} = 0$ **(5.28)**
$\sum_{l=0}^{K} \left\{ \frac{\Delta x_i}{N_x}(\gamma_{i,k}^{0,l}\psi_l, \psi_m)_{\Omega_e} + ((c_{i+1/2,k}^{(a,l)} - c_{i-1/2,k}^{(a,l)})\psi_l, \psi_m)_{\Omega_e} = 0 \right\}$ **(5.29)**
$\sum_{l=0}^{K} \left\{ \frac{1}{N_z}(\chi_{i,k}^{(0,l)}\psi_l, \psi_m)_{\Omega_e} - \left(c_{i,k}^{0,l}\psi_l, \frac{d\psi_m}{dz}\right)_{\Omega_e} + c_{i,k+1/2}^{(0,a)}\psi_l^- - c_{i,k-1/2}^{(0,a)}\psi_l^- = 0 \right\}$ **(5.30)**

Table 5.1 Weak equations for the z direction semi-discretization

$j = 1$

$$\sum_{l=0}^{K}\left\{\frac{\Delta x_i}{3}\frac{d}{dt}(c_{i,k}^{(1,l)}\psi_l,\psi_m)_{\Omega_e}-\right.$$

$$\left((u_{i-1/2,k-1/2}\phi_0+u_{i-1/2,k+1/2}\phi_1)\left(c_{i,k}^{0,l}-\frac{1}{3}c_{i,k}^{1,l}\right)\psi_l,\psi_m\right)_{\Omega_e}-$$

$$\left((u_{i+1/2,k-1/2}\phi_0+u_{i+1/2,k+1/2}\phi_1)\left(c_{i,k}^{(0,l)}+\frac{1}{3}c_{i,k}^{(1,l)}\right)\psi_l,\psi_m\right)_{\Omega_e}-(2\gamma_{i,k}^{(0,l)}\psi_l,\psi_m)_{\Omega_e}+$$

$$((u_{i+1/2,k-1/2}\phi_0+u_{i+1/2,k+1/2}\phi_1)c_{i+1/2,k}^{(u,l)}\psi_l+\gamma_{i+1/2,k}^{(a,l)}\psi_l,\psi_m)_{\Omega_e}+$$

$$((u_{i-1/2,k-1/2}\phi_0+u_{i-1/2,k+1/2}\phi_1)c_{i-1/2,k}^{(u,l)}\psi_l+\gamma_{i-1/2,k}^{(a,l)}\psi_l,\psi_m)_{\Omega_e}-$$

$$\frac{\Delta x_i}{2}\left(\frac{1}{3}(w_{i-1/2,k-1/2}\phi_0+w_{i-1/2,k+1/2}\phi_1)(-c_{i,k}^{(0,l)}+c_{i,k}^{(1,l)})\psi_l,\frac{d\psi_m}{dz}\right)_{\Omega_e}-$$

$$\frac{\Delta x_i}{2}\left(\frac{1}{3}(w_{i+1/2,k-1/2}\phi_0+w_{i+1/2,k+1/2}\phi_1)(c_{i,k}^{(0,l)}+c_{i,k}^{(1,l)})\psi_l,\frac{d\psi_m}{dz}\right)_{\Omega_e}+$$

$$\frac{\Delta x_i}{2}\left[\frac{1}{3}w_{i-1/2,k+1/2}(-c_{i,k+1/2}^{(0,u)}+c_{i,k+1/2}^{(1,u)})+\frac{1}{3}w_{i+1/2,k+1/2}(c_{i,k+1/2}^{(0,u)}+c_{i,k+1/2}^{(1,u)})\right]\psi_l^-\;-$$

$$\frac{\Delta x_i}{2}\left[\frac{1}{3}w_{i-1/2,k-1/2}(-c_{i,k-1/2}^{(0,u)}+c_{i,k-1/2}^{(1,u)})+\frac{1}{3}w_{i+1/2,k-1/2}(c_{i,k-1/2}^{(0,u)}+c_{i,k-1/2}^{(1,u)})\right]\psi_l^-\;+$$

$$\left.\frac{\Delta x_i}{2}\left[\left(\frac{2}{3}\chi_{i,k}^{(1,l)}\psi_l,\frac{d\psi_m}{dz}\right)_{\Omega_e}+\frac{2}{3}(\chi_{i,k+1/2}^{(1,a)}\psi_l^-\;-\chi_{i,k-1/2}^{(1,a)}\psi_l^-)\right]\right\}\;=\;0 \qquad \textbf{(5.31)}$$

$$\sum_{l=0}^{K}\left\{\frac{\Delta x_i}{3N_x}(\gamma_{i,k}^{(1,l)}\psi_l,\psi_m)_{\Omega_e}+((-2c_{i,k}^{(0,l)}+c_{i+1/2,k}^{(a,l)}+c_{i-1/2,k}^{(a,l)})\psi_l,\psi_m)_{\Omega_e}=0\right\} \qquad \textbf{(5.32)}$$

$$\sum_{l=0}^{K}\left\{\frac{1}{N_z}(\chi_{i,k}^{(1,l)}\psi_l,\psi_m)_{\Omega_e}-\left(c_{i,k}^{1,l}\psi_l,\frac{d\psi_m}{dz}\right)_{\Omega_e}+c_{i,k+1/2}^{(1,a)}\psi_l^-\;-c_{i,k-1/2}^{(1,a)}\psi_l^-\;=0\right\} \qquad \textbf{(5.33)}$$

If linear test functions (K=1) are used in the z direction, then the resulting semi-discrete equations for the four unknowns in each element are given in Table 5.2. The four

Table 5.2 Spatial discretization of the x-z transport equation for an interior element (using linear approximating functions).

($j = 0, m = 0$)
$$\Delta x_i \Delta z_k \frac{d}{dt} c_{i,k}^{(0,0)} + \frac{\Delta z_k}{2}[2\gamma_{i+1/2,k}^{(a,0)} - 2\gamma_{i-1/2,k}^{(a,0)}] + \frac{\Delta x_i}{2}[2\chi_{i,k+1/2}^{(0,a)} - 2\chi_{i,k-1/2}^{(0,a)}] +$$ $$\frac{\Delta z_k}{2}\left[u_{i+1/2,k-1/2}\left(c_{i+1/2,k}^{(u,0)} - \frac{1}{3}c_{i+1/2,k}^{(u,1)}\right) + u_{i+1/2,k+1/2}\left(c_{i+1/2,k}^{(u,0)} + \frac{1}{3}c_{i+1/2,k}^{(u,1)}\right)\right] -$$ $$\frac{\Delta z_k}{2}\left[u_{i-1/2,k-1/2}\left(c_{i-1/2,k}^{(u,0)} - \frac{1}{3}c_{i-1/2,k}^{(u,1)}\right) + u_{i-1/2,k+1/2}\left(c_{i-1/2,k}^{(u,0)} + \frac{1}{3}c_{i-1/2,k}^{(u,1)}\right)\right] +$$ $$\frac{\Delta x_i}{2}\left[w_{i-1/2,k+1/2}\left(c_{i,k+1/2}^{(0,u)} - \frac{1}{3}c_{i,k+1/2}^{(1,u)}\right) + w_{i+1/2,k+1/2}\left(c_{i,k+1/2}^{(0,u)} + \frac{1}{3}c_{i,k+1/2}^{(1,u)}\right)\right] -$$ $$\frac{\Delta x_i}{2}\left[w_{i-1/2,k-1/2}\left(c_{i,k-1/2}^{(0,u)} - \frac{1}{3}c_{i,k-1/2}^{(1,u)}\right) + w_{i+1/2,k-1/2}\left(c_{i,k-1/2}^{(0,u)} + \frac{1}{3}c_{i,k-1/2}^{(1,u)}\right)\right] = 0 \quad \textbf{(5.34)}$$
$$\frac{\Delta x_i}{N_x}\gamma_{i,k}^{(0,0)} + c_{i+1/2,k}^{(a,0)} - c_{i-1/2,k}^{(a,0)} = 0 \quad \textbf{(5.35)}$$
$$\frac{\Delta z_k}{N_z}\chi_{i,k}^{(0,0)} + c_{i,k+1/2}^{(0,a)} - c_{i,k-1/2}^{(0,a)} = 0 \quad \textbf{(5.36)}$$
($j = 0, m = 1$)
$$\Delta x_i \frac{\Delta z_k}{3}\frac{d}{dt} c_{i,k}^{(0,1)} + \frac{\Delta z_k}{2}\left[\frac{2}{3}\gamma_{i+1/2,k}^{(a,1)} + \frac{2}{3}\gamma_{i-1/2,k}^{(a,1)}\right] + \frac{\Delta x_i}{2}[2\chi_{i,k+1/2}^{(0,a)} + 2\chi_{i,k-1/2}^{(0,a)} - 4\chi_{i,k}^{(0,0)}] +$$ $$\frac{\Delta z_k}{6}[u_{i+1/2,k-1/2}(-c_{i+1/2,k}^{(u,0)} + c_{i+1/2,k}^{(u,1)}) + u_{i+1/2,k+1/2}(c_{i+1/2,k}^{(u,0)} + c_{i+1/2,k}^{(u,1)})] -$$ $$\frac{\Delta z_k}{6}[u_{i-1/2,k-1/2}(-c_{i-1/2,k}^{(u,0)} + c_{i-1/2,k}^{(u,1)}) + u_{i-1/2,k+1/2}(c_{i-1/2,k}^{(u,0)} + c_{i-1/2,k}^{(u,1)})] -$$ $$\frac{\Delta x_i}{2}\left[w_{i-1/2,k-1/2}\left(c_{i,k}^{(0,0)} - \frac{1}{3}c_{i,k}^{(0,1)} - \frac{1}{3}c_{i,k}^{(1,0)} + \frac{1}{9}c_{i,k}^{(1,1)}\right)\right] -$$ $$\frac{\Delta x_i}{2}\left[w_{i-1/2,k+1/2}\left(c_{i,k}^{(0,0)} + \frac{1}{3}c_{i,k}^{(0,1)} - \frac{1}{3}c_{i,k}^{(1,0)} - \frac{1}{9}c_{i,k}^{(1,1)}\right)\right] -$$

Table 5.2 Spatial discretization of the *x*-*z* transport equation for an interior element (using linear approximating functions).

$$\frac{\Delta x_i}{2}\left[w_{i+1/2,\,k-1/2}\left(c_{i,k}^{(0,0)}-\frac{1}{3}c_{i,k}^{(0,1)}+\frac{1}{3}c_{i,k}^{(1,0)}-\frac{1}{9}c_{i,k}^{(1,1)}\right)\right]-$$

$$\frac{\Delta x_i}{2}\left[w_{i+1/2,\,k+1/2}\left(c_{i,k}^{(0,0)}+\frac{1}{3}c_{i,k}^{(0,1)}+\frac{1}{3}c_{i,k}^{(1,0)}+\frac{1}{9}c_{i,k}^{(1,1)}\right)\right]+$$

$$\frac{\Delta x_i}{2}\left[w_{i-1/2,\,k+1/2}\left(c_{i,k+1/2}^{(0,u)}-\frac{1}{3}c_{i,k+1/2}^{(1,u)}\right)+w_{i+1/2,\,k+1/2}\left(c_{i,k+1/2}^{(0,u)}+\frac{1}{3}c_{i,k+1/2}^{(1,u)}\right)\right]+$$

$$\frac{\Delta x_i}{2}\left[w_{i-1/2,\,k-1/2}\left(c_{i,k-1/2}^{(0,u)}-\frac{1}{3}c_{i,k-1/2}^{(1,u)}\right)+w_{i+1/2,\,k-1/2}\left(c_{i,k-1/2}^{(0,u)}+\frac{1}{3}c_{i,k-1/2}^{(1,u)}\right)\right]=0 \quad \textbf{(5.37)}$$

$$\frac{\Delta x_i}{N_x}\gamma_{i,k}^{(0,1)}+c_{i+1/2,\,k}^{(a,1)}-c_{i-1/2,\,k}^{(a,1)}=0 \quad \textbf{(5.38)}$$

$$\frac{\Delta z_k}{3N_z}\chi_{i,k}^{(0,1)}-2c_{i,k}^{(0,0)}+c_{i,k+1/2}^{(0,a)}+c_{i,k-1/2}^{(0,a)}=0 \quad \textbf{(5.39)}$$

($j = 1$, $m = 0$)

$$\frac{\Delta x_i}{3}\Delta z_k\frac{d}{dt}c_{i,k}^{(1,0)}+\frac{\Delta z_k}{2}[2\gamma_{i+1/2,\,k}^{(a,0)}+2\gamma_{i-1/2,\,k}^{(a,0)}-4\gamma_{i,k}^{(0,0)}]+\frac{\Delta x_i}{2}\left[\frac{2}{3}\chi_{i,k+1/2}^{(1,a)}+\frac{2}{3}\chi_{i,k-1/2}^{(1,a)}\right]-$$

$$\frac{\Delta z_k}{2}\left[u_{i-1/2,\,k-1/2}\left(c_{i,k}^{(0,0)}-\frac{1}{3}c_{i,k}^{(0,1)}-\frac{1}{3}c_{i,k}^{(1,0)}+\frac{1}{9}c_{i,k}^{(1,1)}\right)\right]-$$

$$\frac{\Delta z_k}{2}\left[u_{i-1/2,\,k+1/2}\left(c_{i,k}^{(0,0)}+\frac{1}{3}c_{i,k}^{(0,1)}-\frac{1}{3}c_{i,k}^{(1,0)}-\frac{1}{9}c_{i,k}^{(1,1)}\right)\right]-$$

$$\frac{\Delta z_k}{2}\left[u_{i+1/2,\,k-1/2}\left(c_{i,k}^{(0,0)}-\frac{1}{3}c_{i,k}^{(0,1)}+\frac{1}{3}c_{i,k}^{(1,0)}-\frac{1}{9}c_{i,k}^{(1,1)}\right)\right]-$$

$$\frac{\Delta z_k}{2}\left[u_{i+1/2,\,k+1/2}\left(c_{i,k}^{(0,0)}+\frac{1}{3}c_{i,k}^{(0,1)}+\frac{1}{3}c_{i,k}^{(1,0)}+\frac{1}{9}c_{i,k}^{(1,1)}\right)\right]+$$

$$\frac{\Delta z_k}{2}\left[u_{i+1/2,\,k-1/2}\left(c_{i+1/2,\,k}^{(u,0)}-\frac{1}{3}c_{i+1/2,\,k}^{(u,1)}\right)+u_{i+1/2,\,k+1/2}\left(c_{i+1/2,\,k}^{(u,0)}+\frac{1}{3}c_{i+1/2,\,k}^{(u,1)}\right)\right]+$$

$$\frac{\Delta z_k}{2}\left[u_{i-1/2,\,k-1/2}\left(c_{i-1/2,\,k}^{(u,0)}-\frac{1}{3}c_{i-1/2,\,k}^{(u,1)}\right)+u_{i-1/2,\,k+1/2}\left(c_{i-1/2,\,k}^{(u,0)}+\frac{1}{3}c_{i-1/2,\,k}^{(u,1)}\right)\right]+$$

$$\frac{\Delta x_i}{6}[w_{i-1/2,\,k+1/2}(-c_{i,k+1/2}^{(0,u)}+c_{i,k+1/2}^{(1,u)})+w_{i+1/2,\,k+1/2}(c_{i,k+1/2}^{(0,u)}+c_{i,k+1/2}^{(1,u)})]-$$

Table 5.2 Spatial discretization of the *x*-*z* transport equation for an interior element (using linear approximating functions).

$\frac{\Delta x_i}{6}[w_{i-1/2,\,k-1/2}(-c^{(0,u)}_{i,\,k-1/2}+c^{(1,u)}_{i,\,k-1/2})+w_{i+1/2,\,k-1/2}(c^{(0,u)}_{i,\,k-1/2}+c^{(1,u)}_{i,\,k-1/2})]=0$ **(5.40)**
$\frac{\Delta x_i}{3N_x}\gamma^{(1,0)}_{i,\,k}-2c^{(0,0)}_{i,\,k}+c^{(a,0)}_{i+1/2,\,k}+c^{(a,0)}_{i-1/2,\,k}=0$ **(5.41)**
$\frac{\Delta z_k}{N_z}\chi^{(1,0)}_{i,\,k}+c^{(1,a)}_{i,\,k+1/2}-c^{(1,a)}_{i,\,k-1/2}=0$ **(5.42)**
($j=1, m=1$)
$\frac{\Delta x_i}{3}\frac{\Delta z_k}{3}\frac{d}{dt}c^{(1,1)}_{i,\,k}+\frac{\Delta z_k}{2}\left[\frac{2}{3}\gamma^{(a,1)}_{i+1/2,\,k}+\frac{2}{3}\gamma^{(a,1)}_{i-1/2,\,k}-\frac{4}{3}\gamma^{(0,1)}_{i,\,k}\right]+$
$\frac{\Delta x_i}{2}\left[\frac{2}{3}\chi^{(1,a)}_{i,\,k+1/2}+\frac{2}{3}\chi^{(1,a)}_{i,\,k-1/2}-\frac{4}{3}\chi^{(1,0)}_{i,\,k}\right]-$
$\frac{\Delta z_k}{6}\left[u_{i-1/2,\,k-1/2}\left(-c^{(0,0)}_{i,\,k}+c^{(0,1)}_{i,\,k}+\frac{1}{3}c^{(1,0)}_{i,\,k}-\frac{1}{3}c^{(1,1)}_{i,\,k}\right)\right]-$
$\frac{\Delta z_k}{6}\left[u_{i-1/2,\,k+1/2}\left(c^{(0,0)}_{i,\,k}+c^{(0,1)}_{i,\,k}-\frac{1}{3}c^{(1,0)}_{i,\,k}-\frac{1}{3}c^{(1,1)}_{i,\,k}\right)\right]-$
$\frac{\Delta z_k}{6}\left[u_{i+1/2,\,k-1/2}\left(-c^{(0,0)}_{i,\,k}+c^{(0,1)}_{i,\,k}-\frac{1}{3}c^{(1,0)}_{i,\,k}+\frac{1}{3}c^{(1,1)}_{i,\,k}\right)\right]-$
$\frac{\Delta z_k}{6}\left[u_{i+1/2,\,k+1/2}\left(c^{(0,0)}_{i,\,k}+c^{(0,1)}_{i,\,k}+\frac{1}{3}c^{(1,0)}_{i,\,k}+\frac{1}{3}c^{(1,1)}_{i,\,k}\right)\right]+$
$\frac{\Delta z_k}{6}[u_{i+1/2,\,k-1/2}(-c^{(u,0)}_{i+1/2,\,k}+c^{(u,1)}_{i+1/2,\,k})+u_{i+1/2,\,k+1/2}(c^{(u,0)}_{i+1/2,\,k}+c^{(u,1)}_{i+1/2,\,k})]+$
$\frac{\Delta z_k}{6}[u_{i-1/2,\,k-1/2}(-c^{(u,0)}_{i-1/2,\,k}+c^{(u,1)}_{i-1/2,\,k})+u_{i-1/2,\,k+1/2}(c^{(u,0)}_{i-1/2,\,k}+c^{(u,1)}_{i-1/2,\,k})]+$
$\frac{\Delta x_i}{6}[w_{i-1/2,\,k+1/2}(-c^{(0,u)}_{i,\,k+1/2}+c^{(1,u)}_{i,\,k+1/2})+w_{i+1/2,\,k+1/2}(c^{(0,u)}_{i,\,k+1/2}+c^{(1,u)}_{i,\,k+1/2})]+$
$\frac{\Delta x_i}{6}[w_{i-1/2,\,k-1/2}(-c^{(0,u)}_{i,\,k-1/2}+c^{(1,u)}_{i,\,k-1/2})+w_{i+1/2,\,k-1/2}(c^{(0,u)}_{i,\,k-1/2}+c^{(1,u)}_{i,\,k-1/2})]-$
$\frac{\Delta x_i}{6}\left[w_{i-1/2,\,k-1/2}\left(-c^{(0,0)}_{i,\,k}+\frac{1}{3}c^{(0,1)}_{i,\,k}+c^{(1,0)}_{i,\,k}-\frac{1}{3}c^{(1,1)}_{i,\,k}\right)\right]-$

Table 5.2 Spatial discretization of the *x*-*z* transport equation for an interior element (using linear approximating functions).

$$\frac{\Delta x_i}{6}\left[w_{i-1/2,\,k+1/2}\left(-c_{i,k}^{(0,0)}-\frac{1}{3}c_{i,k}^{(0,1)}+c_{i,k}^{(1,0)}+\frac{1}{3}c_{i,k}^{(1,1)}\right)\right]-$$

$$\frac{\Delta x_i}{6}\left[w_{i+1/2,\,k-1/2}\left(c_{i,k}^{(0,0)}-\frac{1}{3}c_{i,k}^{(0,1)}+c_{i,k}^{(1,0)}-\frac{1}{3}c_{i,k}^{(1,1)}\right)\right]-$$

$$\frac{\Delta x_i}{6}\left[w_{i+1/2,\,k+1/2}\left(c_{i,k}^{(0,0)}+\frac{1}{3}c_{i,k}^{(0,1)}+c_{i,k}^{(1,0)}+\frac{1}{3}c_{i,k}^{(1,1)}\right)\right]=0 \qquad \textbf{(5.43)}$$

$$\frac{\Delta x_i}{3N_x}\gamma_{i,k}^{(1,1)}-2c_{i,k}^{(0,1)}+c_{i+1/2,\,k}^{(a,1)}+c_{i-1/2,\,k}^{(a,1)}=0 \qquad \textbf{(5.44)}$$

$$\frac{\Delta z_k}{3N_z}\chi_{i,k}^{(1,1)}-2c_{i,k}^{(1,0)}+c_{i,\,k+1/2}^{(1,a)}+c_{i,\,k-1/2}^{(1,a)}=0 \qquad \textbf{(5.45)}$$

degrees of freedom can be interpreted as the element average, the *z* slope, the *x* slope and the *x*-*z* slope, respectively. Note that these differ from the standard bi-linear forms since we are not solving Equation (5.1) in a traditional 2D sense, but as a series of 1D discretizations.

Equations (5.34), (5.37), (5.40), and (5.43) are then discretized in time to solve for the four unknowns $c_{i,k}^{(0,0)}$, $c_{i,k}^{(0,1)}$, $c_{i,k}^{(1,0)}$ and $c_{i,k}^{(1,1)}$. For this study, a simple explicit or forward Euler time discretization is used, which allows the four equations to be solved independently of each other. Note that the auxiliary variables $\gamma_{i,k}^{(j,m)}$ and $\chi_{i,k}^{(j,m)}$ can be eliminated by directly substituting Equations (5.35), (5.36), (5.38), (5.39), (5.41), (5.42), (5.44) and (5.45) into Equations (5.34), (5.37), (5.40), and (5.43).

Note also that due to the averaging in the auxiliary variables, the solution on each interior element depends upon its neighbors and their neighbors; thus creating a nine element stencil, as shown in Figure 5.4. On the boundaries, this is reduced according to the replacements discussed in the next section; but this rather large stencil requires that the neighbors of the boundary elements will also be affected by the BCs, requiring special care on these elements as well. However, as long as an "average" value is defined on all of the

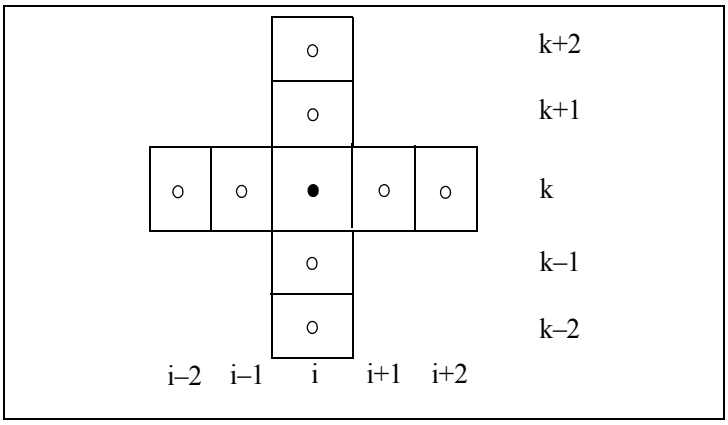


Figure 5.4 Discretization stencil for element (i, k).

external boundary interfaces, the neighbors of the boundary elements will be well-defined. Note also that since only a single value exists at these interfaces, the average value is the same as the upwind value.

5.3.3 Boundary conditions

Herein, three types of boundary conditions (BCs) are examined for the above LDG transport algorithm: (I) Dirichlet, (II) natural or inflow/outflow, and (III) no total flux. Below, the basis for each of these boundary types is presented. For brevity, it is easier to explain each method in the framework of the *x* direction equations (5.17-5.18) rather than the full *x*-*z* equations presented in Tables 5.1 and 5.2. However, the same premises apply to the *z* direction boundaries.

Type I: Dirichlet

Specification of the dependent variable at the boundary is referred to as a first-type or Dirichlet boundary condition. Dirichlet boundary conditions are often used for algorithm testing and take the form $c = \tilde{c}$ on the boundary. This can be enforced by setting $\gamma^a = \gamma^-$ in Equation (5.17) and $c^a = \tilde{c}$ in Equation (5.18) when integrating over the boundary. Additionally, in the definition of c^u in Equation (5.15), set $c^+ = \tilde{c}$.

Type II: natural or inflow/outflow

What is referred to as a type II, or natural BC, in this work is not strictly a mathematical Type II BC, since inflow conditions also place a restriction on the concentration itself (i.e., for outflow conditions it is a traditional Type II BC, but for inflow conditions it is a Type III BC). In practice, flux or inflow/outflow conditions are most natural to use for actual applications. An inflow condition is imposed on the boundary when $u \bullet n_e < 0$ and an outflow condition is specified otherwise. On the inflow boundaries set

$$\langle (uc^u + \gamma^a) \cdot n_e, \varphi^I \rangle_{\partial\Omega_e} = \langle uc_I(t) \cdot n_e, \varphi^I \rangle_{\partial\Omega_e} \quad \textbf{(5.46)}$$

in Equation (5.17), where $c_I(t)$ is the imposed inflow concentration and u is the specified or computed boundary velocity from the hydrodynamics. On the outflow boundaries we set

$$\langle \gamma^a \cdot n_e, \varphi^I \rangle_{\partial\Omega_e} = 0 \quad \textbf{(5.47)}$$

in Equation (5.17) and set $c^a = c^-$ in Equation (5.18).

Type III: no total flux (Robin)

Specification of a linear combination of the dependent variable and its normal gradient along a boundary is referred to as a third-type or Robin boundary condition. A no total flux boundary condition is required in applications where no advective or diffusive flux is desired across the external boundaries, i.e., a completely closed domain. This is taken care of naturally with Type II BCs in the x direction when land boundaries are specified on both ends of the domain for the hydrodynamics. This locks the normal boundary velocity to 0.0 m/s and automatically removes the advective flux in the transport. (When $u \bullet n_e = 0$ Type II normal boundaries automatically become no-flow boundaries since the diffusive flux has already been removed.) Thus, the combination of land boundaries for the hydrodynamics and Type II BCs for the transport will ensure a Type III transport boundary condition for the horizontal boundaries.

However, the vertical velocities, w, are most often not equivalently zero at the sea surface where the top of the domain is constantly moving. Here, additional restraints must be made within the transport algorithm to ensure a no total flux boundary condition. The above modifications made for Type II BCs applied to the z boundaries will remove the diffusive flux, but further care must be taken to ensure that the advective flux is also zero. This can be accomplished by temporarily setting w = 0.0 (only within the transport subroutine - does not change hydrodynamic values) on the boundaries in order to remove the advective fluxes.

5.3.4 Post-processing

For added stability in the algorithm, slope limiting is employed on the three slope degrees of freedom, $c^{(1,0)}$, $c^{(0,1)}$ and $c^{(1,1)}$. After calculating the unknowns for all of the elements in the domain, the slope values are compared and limited as needed. Based on experience from other DG applications, the minmod limiter, which is defined as

$$\text{minmod}(m_1, m_2, m_3) = \begin{cases} s_1 \text{MIN}(|m_1|, |m_2|, |m_3|), \\ \quad \text{when } s_1 = s_2 = s_3 \\ \quad \text{else} \qquad\qquad 0.0 \end{cases} \tag{5.48}$$

where m_1 is the computed slope $c_{i,k}^{(1,0)}$ ($c_{i,k}^{(0,1)}$) for the x (z) direction, m_2 is the left (bottom) slope, m_3 is the right (top) slope and s_1, s_2, s_3 are the signs of their respective slopes, is used in this study. The comparison slopes, m_2 and m_3, are calculated as

$$\text{left} = c_{i,k}^{(0,0)} - c_{i-1,k}^{(0,0)} \tag{5.49}$$

$$\text{right} = c_{i+1,k}^{(0,0)} - c_{i,k}^{(0,0)} \tag{5.50}$$

$$\text{bottom} = c_{i,k}^{(0,0)} - c_{i,k-1}^{(0,0)} \tag{5.51}$$

$$\text{top} = c_{i,k+1}^{(0,0)} - c_{i,k}^{(0,0)}. \tag{5.52}$$

On the domain boundaries, the comparison is made only between the computed slope value and the nearest neighbor in the coordinate direction. Thus, for the left and bottom boundaries the limiter is called with m_1 and m_3 as inputs, while for the right and top boundaries the limiter is called with m_1 and m_2 as inputs. After limiting the x and z slopes, the x-z slope is then limited on each element according to the following rule:

$$\text{IF}(c_{i,k}^{(1,0)} = 0.0 \text{ or } c_{i,k}^{(0,1)} = 0.0) \text{ THEN } c_{i,k}^{(1,1)} = 0.0 \tag{5.53}$$

which sets the x-z slope equal to zero if either of the x or z slopes are zero. As will be seen in §5.5, the x-z slope can be interpreted as the z slope of the x slope or vice versa. Therefore, if either slope is zero, then the slope of that slope is also zero.

With the new limited slope values, the interface values are then calculated for each element, the upwind value is determined, and the average value is calculated in preparation for the next time step. On each element, the *interior* interface values for the two degrees of freedom (0/1) are calculated as

$$R_i(0/1) = c_{i,k}^{(0,0/1)} - c_{i,k}^{(1,0/1)} \tag{5.54}$$

$$L_i(0/1) = c_{i,k}^{(0,0/1)} + c_{i,k}^{(1,0/1)} \tag{5.55}$$

$$T_k(0/1) = c_{i,k}^{(0/1,0)} - c_{i,k}^{(0/1,1)} \tag{5.56}$$

$$B_k(0/1) = c_{i,k}^{(0/1,0)} + c_{i,k}^{(0/1,1)}, \tag{5.57}$$

where R, L, T, B represent right, left, top and bottom, respectively. Then the upwind and average values are calculated according to Equations (5.15) and (5.16) using these interface values. The interface notation for an interior element is shown in Figure 5.5. On the domain

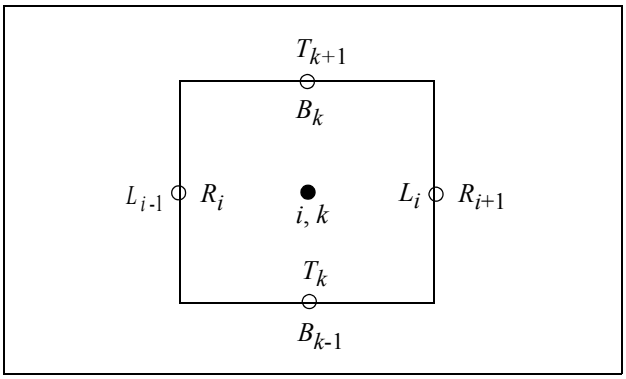


Figure 5.5 Interface notation for interior element (i,k).

boundaries, the exterior state values do not exist and the equations are modified as described in §5.3.3.

5.4 Interpolation to CG grid space

Now it is necessary to return to the larger framework in which the transport equation is being used. In order to update the density field for a prognostic baroclinic simulation, the discontinuous transport variables must be communicated to the continuous hydrodynamic grid space. (The above derivations are valid for transport of any "dissolved" species and can be used to transport both the salinity and potential temperature needed in a full prognostic algorithm. Herein, only the transport of the salinity variable is examined, but the same premise is valid for the potential temperature.)

In this study, two options for interpolating the DG salinity values to the CG grid have been examined. The basic premise of each of these is presented below. But first the grid notation that will be used for each of the interpolation methods is summarized. Note that each interior CG node is surrounded by four DG elements, as shown in Figure 5.6, and each boundary (corner) CG node is surrounded by two (one) DG element(s).

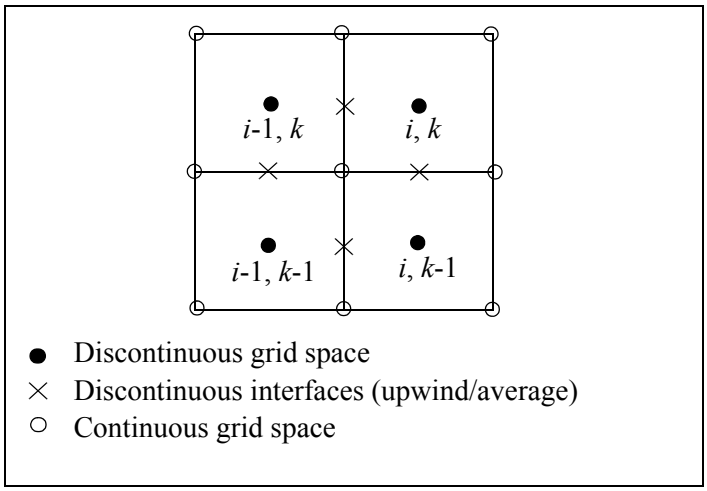


Figure 5.6 Grid space notation for interpolation of CG nodal values from DG element values.

5.4.1 L_2 interpolation

The first attempt to translate the DG state variables to a CG grid space was to use an L_2 integration technique. Following the derivation of the discrete transport equation, this derivation is also separated into two one-dimensional steps; first integrating in the x direction and then integrating the resulting equations in the z direction. The following notation will be used in the derivation:

$c_{i,k}^{(j,m)}$ = element centered DG solution

$c_{i,k}^{CG}$ = projected CG solution

ω = continuous linear Lagrange basis functions

Ω_x = x domain

Ω_z = z domain

The interpolated CG values in the x direction are defined to be the solution of the following weak formulation

$$\int_{\Omega_x} \hat{c}_{i,k}^{CG} \omega_i dx = \int_{\Omega_x} \hat{c}_{i,k}^{(j,m)} \omega_i dx \tag{5.58}$$

where the hat notation indicates that the coefficients are still functions of z and the domain integration can be broken into a sum of element integrations over the entire domain.

If linear Lagrange bases are used for the CG variables and the same linear Legendre bases described in §5.3.1 are used for the DG values, then the approximation becomes

$$\sum_{e=1}^{nx} ((\hat{c}_{i-1/2,k}^{CG} \omega_0 + \hat{c}_{i+1/2,k}^{CG} \omega_1)\omega_j \partial x)_{\Omega_e} =$$

$$\sum_{e=1}^{nx} ((\hat{c}_{i,k}^{(0,m)} \varphi_0 + \hat{c}_{i,k}^{(1,m)} \varphi_1)\omega_j \partial x)_{\Omega_e} \tag{5.59}$$

where nx is the number of elements in the x direction. After lumping the left-hand side of the equation to avoid solving simultaneous equations, the master element becomes

$$\Delta x_i \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} \hat{c}_{i-1/2,k}^{CG} \\ \hat{c}_{i+1/2,k}^{CG} \end{bmatrix} = \Delta x_i \begin{bmatrix} 0.5 & -1 \\ 0.5 & 1 \end{bmatrix} \begin{bmatrix} \hat{c}_{i,k}^{(0,m)} \\ \hat{c}_{i,k}^{(1,m)} \end{bmatrix} \tag{5.60}$$

When element contributions are summed this gives a rectangular matrix on the right-hand side, which is due to the discontinuities at the boundaries. Based on location, there are three equations to carry forward from the x direction L_2 interpolation to the z direction integrations.

Left boundary:

$$\frac{\Delta x_1}{2} \hat{c}_{1/2,k}^{CG} = \Delta x_1 \left(\frac{1}{2} \hat{c}_{1,k}^{(0,m)} - \hat{c}_{1,k}^{(1,m)} \right) \tag{5.61}$$

Interior nodes:

$$\frac{1}{2}(\Delta x_{i-1}+\Delta x_i)\hat{c}^{CG}_{i-1/2,\,k} = \Delta x_{i-1}\left(\frac{1}{2}\hat{c}^{(0,\,m)}_{i-1,\,k} + \hat{c}^{(1,\,m)}_{i-1,\,k}\right) + \Delta x_i\left(\frac{1}{2}\hat{c}^{(0,\,m)}_{i,\,k} - \hat{c}^{(1,\,m)}_{i,\,k}\right) \tag{5.62}$$

Right boundary:

$$\frac{\Delta x_{nx}}{2}\hat{c}^{CG}_{nx+1/2,\,k} = \frac{\Delta x_{nx}}{2}\hat{c}^{(0,\,m)}_{nx,\,k} - \Delta x_{nx}\hat{c}^{(1,\,m)}_{nx,\,k} \tag{5.63}$$

To complete the L_2 interpolation, each of these must now be integrated in the z direction. The general weak form follows the x integrations and is

$$\int_{\Omega_z} c^{CG}_{i,\,k}\omega_k dz = \int_{\Omega_z} c^{(j,\,m)}_{i,\,k}\omega_k dz \tag{5.64}$$

where the hat notation has been removed to indicate that the coefficients are now functions of z alone. Again, these domain integrations can be broken into a sum of element integrations over the entire domain. Linear Lagrange bases are used for the CG variables and the linear Legendre bases described in §5.3.2 are used for the DG variables.

Continuing from Equation (5.61), the integration in the z direction for the left boundary becomes

$$\sum_{e=1}^{nz}\left\{\frac{1}{2}((c^{CG}_{1/2,\,k-1/2}\omega_0 + c^{CG}_{1/2,\,k+1/2}\omega_1)\omega_k dz)_{\Omega_e} = \right.$$

$$\left.\frac{1}{2}((c^{(0,\,0)}_{1,\,k}\psi_0 + c^{(0,\,1)}_{1,\,k}\psi_1)\omega_k dz)_{\Omega_e} - ((c^{(1,\,0)}_{1,\,k}\psi_0 + c^{(1,\,1)}_{1,\,k}\psi_1)\omega_k dz)_{\Omega_e}\right\} \tag{5.65}$$

where nz is the number of elements in the z direction. After lumping the left-hand side, the master element for the left boundary is

$$\Delta z_k \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix} \begin{bmatrix} c^{CG}_{1/2,\,k-1/2} \\ c^{CG}_{1/2,\,k+1/2} \end{bmatrix} = \Delta z_k \begin{bmatrix} 0.25 & -0.5 & -0.5 & 1 \\ 0.25 & 0.5 & -0.5 & -1 \end{bmatrix} \begin{bmatrix} c^{(0,0)}_{1,\,k} \\ c^{(0,1)}_{1,\,k} \\ c^{(1,0)}_{1,\,k} \\ c^{(1,1)}_{1,\,k} \end{bmatrix} \tag{5.66}$$

When the element contributions are summed up, there are three formulations for the interpolated CG nodal values along the left boundary.

Left-bottom corner:

$$\frac{1}{4} c^{CG}_{1/2,\,1/2} = \frac{1}{4} c^{(0,0)}_{1,1} - \frac{1}{2} c^{(0,1)}_{1,1} - \frac{1}{2} c^{(1,0)}_{1,1} + c^{(1,1)}_{1,1} \tag{5.67}$$

Left-interior boundary:

$$\frac{1}{4}(\Delta z_{k-1} + \Delta z_k) c^{CG}_{1/2,\,k-1/2} = \Delta z_{k-1}\left(\frac{1}{4} c^{(0,0)}_{1,\,k-1} + \frac{1}{2} c^{(0,1)}_{1,\,k-1} - \frac{1}{2} c^{(1,0)}_{1,\,k-1} - c^{(1,1)}_{1,\,k-1}\right) +$$

$$\Delta z_k\left(\frac{1}{4} c^{(0,0)}_{1,\,k} - \frac{1}{2} c^{(0,1)}_{1,\,k} - \frac{1}{2} c^{(1,0)}_{1,\,k} + c^{(1,1)}_{1,\,k}\right) \tag{5.68}$$

Left-top corner:

$$\frac{1}{4} c^{CG}_{1/2,\,nz+1/2} = \frac{1}{4} c^{(0,0)}_{1,\,nz} - \frac{1}{2} c^{(0,1)}_{1,\,nz} - \left(\frac{1}{2} c^{(1,0)}_{1,\,nz} + c^{(1,1)}_{1,\,nz}\right) \tag{5.69}$$

Continuing from Equation (5.62), the integration in the z direction for the interior nodes gives

$$\sum_{e=1}^{nz} \left\{ \frac{1}{2}(\Delta x_{i-1} + \Delta x_i)((c^{CG}_{i-1/2,\,k-1/2}\omega_0 + c^{CG}_{i-1/2,\,k+1/2}\omega_1)\omega_k dz)_{\Omega_e} =$$

$$\frac{\Delta x_{i-1}}{2}((c^{(0,0)}_{i-1,\,k}\psi_0 + c^{(0,1)}_{i-1,\,k}\psi_1)\omega_k dz)_{\Omega_e} + \Delta x_{i-1}((c^{(1,0)}_{i-1,\,k}\psi_0 + c^{(1,1)}_{i-1,\,k}\psi_1)\omega_k dz)_{\Omega_e} +$$

$$\frac{\Delta x_i}{2}((c_{i,k}^{(0,0)}\psi_0 + c_{i,k}^{(0,1)}\psi_1)\omega_k dz)_{\Omega_e} - \Delta x_i((c_{i,k}^{(1,0)}\psi_0 + c_{i,k}^{(1,1)}\psi_1)\omega_k dz)_{\Omega_e}\Big\} \quad \textbf{(5.70)}$$

When the left-hand side is lumped, the master element for interior nodes is given as

$$(\Delta x_{i-1} + \Delta x_i)\frac{\Delta z_k}{2}\begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}\begin{bmatrix} c_{i-1/2,\,k-1/2}^{CG} \\ c_{i-1/2,\,k+1/2}^{CG} \end{bmatrix} =$$

$$\Delta z_k \begin{bmatrix} 0.25 & -0.5 & 0.5 & -1 & 0.25 & -0.5 & -0.5 & 1 \\ 0.25 & 0.5 & 0.5 & 1 & 0.25 & 0.5 & -0.5 & -1 \end{bmatrix} C_{\Delta x} \quad \textbf{(5.71)}$$

where

$$C_{\Delta x} = \begin{bmatrix} \Delta x_{i-1} c_{i-1,\,k}^{(0,0)} \\ \Delta x_{i-1} c_{i-1,\,k}^{(0,1)} \\ \Delta x_{i-1} c_{i-1,\,k}^{(1,0)} \\ \Delta x_{i-1} c_{i-1,\,k}^{(1,1)} \\ \Delta x_i\, c_{i,\,k}^{(0,0)} \\ \Delta x_i\, c_{i,\,k}^{(0,1)} \\ \Delta x_i\, c_{i,\,k}^{(1,0)} \\ \Delta x_i\, c_{i,\,k}^{(1,1)} \end{bmatrix}$$

Again summing the element contributions gives three formulations for the interpolated CG nodal values in the interior (away from the left and right boundaries).

<u>Bottom-interior boundary:</u>

$$\frac{\Delta x_{i-1}+\Delta x_i}{4} c_{i-1/2,\,1/2}^{CG} = \Delta x_{i-1}\left(\frac{1}{4}c_{i-1,\,1}^{(0,0)} - \frac{1}{2}c_{i-1,\,1}^{(0,1)} + \frac{1}{2}c_{i-1,\,1}^{(1,0)} - c_{i-1,\,1}^{(1,1)}\right) +$$

$$\Delta x_i\left(\frac{1}{4}c_{i,\,1}^{(0,0)} - \frac{1}{2}c_{i,\,1}^{(0,1)} - \frac{1}{2}c_{i,\,1}^{(1,0)} + c_{i,\,1}^{(1,1)}\right) \quad \textbf{(5.72)}$$

Interior nodes:

$$\frac{\Delta x_{i-1}+\Delta x_i}{4}(\Delta z_{k-1}+\Delta z_k)c^{CG}_{i-1/2,\,k-1/2} =$$

$$\Delta z_{k-1}\left(\Delta x_{i-1}\left(\frac{1}{4}c^{(0,0)}_{i-1,\,k-1}+\frac{1}{2}c^{(0,1)}_{i-1,\,k-1}+\frac{1}{2}c^{(1,0)}_{i-1,\,k-1}+c^{(1,1)}_{i-1,\,k-1}\right)+\right.$$

$$\left.\Delta x_i\left(\frac{1}{4}c^{(0,0)}_{i,\,k-1}+\frac{1}{2}c^{(0,1)}_{i,\,k-1}-\frac{1}{2}c^{(1,0)}_{i,\,k-1}-c^{(1,1)}_{i,\,k-1}\right)\right)+\Delta z_k\left(\Delta x_{i-1}\left(\frac{1}{4}c^{(0,0)}_{i-1,\,k}-\right.\right.$$

$$\left.\left.\frac{1}{2}c^{(0,1)}_{i-1,\,k}+\frac{1}{2}c^{(1,0)}_{i-1,\,k}-c^{(1,1)}_{i-1,\,k}\right)+\Delta x_i\left(\frac{1}{4}c^{(0,0)}_{i,\,k}-\frac{1}{2}c^{(0,1)}_{i,\,k}-\frac{1}{2}c^{(1,0)}_{i,\,k}+c^{(1,1)}_{i,\,k}\right)\right) \qquad \textbf{(5.73)}$$

Top-interior boundary:

$$\frac{\Delta x_{i-1}+\Delta x_i}{4}c^{CG}_{i-1/2,\,nz+1/2} = \Delta x_{i-1}\left(\frac{1}{4}c^{(0,0)}_{i-1,\,nz}+\frac{1}{2}c^{(0,1)}_{i-1,\,nz}+\frac{1}{2}c^{(1,0)}_{i-1,\,nz}+c^{(1,1)}_{i-1,\,nz}\right)+$$

$$\Delta x_i\left(\frac{1}{4}c^{(0,0)}_{i,\,nz}+\frac{1}{2}c^{(0,1)}_{i,\,nz}-\frac{1}{2}c^{(1,0)}_{i,\,nz}-c^{(1,1)}_{i,\,nz}\right) \qquad \textbf{(5.74)}$$

Continuing from Equation (5.63), the integration in the z direction for the right boundary is given as

$$\sum_{e-1}^{nz}\left\{\frac{1}{2}\left((c^{CG}_{nx+1/2,\,k-1/2}\omega_0+c^{CG}_{nx+1/2,\,k+1/2}\omega_0)\omega_k dz\right)_{\Omega_e}=\right.$$

$$\left.\frac{1}{2}((c^{(0,0)}_{nx,\,k}\psi_0+c^{(0,1)}_{nx,\,k}\psi_1)\omega_k dz)_{\Omega_e}-((c^{(1,0)}_{nx,\,k}\psi_0+c^{(1,1)}_{nx,\,k}\psi_1)\omega_k dz)_{\Omega_e}\right\} \qquad \textbf{(5.75)}$$

After lumping, the master element for the right boundary is

$$\Delta z_k \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix} \begin{bmatrix} c^{CG}_{nx+1/2,\,k-1/2} \\ c^{CG}_{nx+1/2,\,k+1/2} \end{bmatrix} = \Delta z_k \begin{bmatrix} 0.25 & -0.5 & 0.5 & -1 \\ 0.25 & 0.5 & 0.5 & 1 \end{bmatrix} \begin{bmatrix} c^{(0,\,0)}_{nx,\,k} \\ c^{(0,\,1)}_{nx,\,k} \\ c^{(1,\,0)}_{nx,\,k} \\ c^{(1,\,1)}_{nx,\,k} \end{bmatrix} \tag{5.76}$$

After summing the element contributions, there are three formulas for the interpolated CG nodal values along the right boundary.

Right-bottom corner:

$$\frac{1}{4} c^{CG}_{nx+1/2,\,1/2} = \frac{1}{4} c^{(0,0)}_{nx,1} - \frac{1}{2} c^{(0,1)}_{nx,1} + \frac{1}{2} c^{(1,0)}_{nx,1} - c^{(1,1)}_{nx,1} \tag{5.77}$$

Right-interior boundary:

$$\frac{1}{4}(\Delta z_{k-1} + \Delta z_k) c^{CG}_{nx+1/2,\,k-1/2} = \Delta z_{k-1}\left(\frac{1}{4} c^{(0,0)}_{nx,\,k-1} + \frac{1}{2} c^{(0,1)}_{nx,\,k-1} + \frac{1}{2} c^{(1,0)}_{nx,\,k-1} + c^{(1,1)}_{nx,\,k-1}\right) +$$

$$\Delta z_k\left(\frac{1}{4} c^{(0,0)}_{nx,\,k} - \frac{1}{2} c^{(0,1)}_{nx,\,k} + \frac{1}{2} c^{(1,0)}_{nx,\,k} - c^{(1,1)}_{nx,\,k}\right) \tag{5.78}$$

Right-top corner:

$$\frac{1}{4} c^{CG}_{nx+1/2,\,nz+1/2} = \frac{1}{4} c^{(0,0)}_{nx,\,nz} + \frac{1}{2} c^{(0,1)}_{nx,\,nz} + \frac{1}{2} c^{(1,0)}_{nx,\,nz} + c^{(1,1)}_{nx,\,nz} \tag{5.79}$$

5.4.2 Averaging from adjacent interfaces

In order to minimize the computational effort of the algorithm, another method, which uses the DG interface data was examined. This interface information is already computed during the DG calculations, and its use to translate to the CG grid would eliminate a costly series of integrations over the entire domain, as is done in the L_2

interpolation. This averaging technique will be referred to as interface averaging in all subsequent discussions.

As part of the DG calculations, upwind and average values are assigned to each element interface throughout the domain (refer to red dots in Figure 5.6). Each interior CG node is surrounded by four elements and thus four interfaces, while each CG boundary (corner) node is surrounded by two (one) element(s) with three (two) adjacent interfaces. For the interior and corner CG nodes, information from all adjacent interfaces is averaged to arrive at the translated CG value. However, for the non-corner boundary CG nodes, only the interfaces that are located on the boundary are averaged to determine the translated CG value (perpendicular interfaces are not used). This results in interface averaging formulas for the various CG node locations.

Left-bottom corner:

$$c^{CG}_{1/2,\,1/2} = \frac{1}{2}(c^{(a,\,0)}_{1/2,\,1} + c^{(0,\,a)}_{1,\,1/2}) \tag{5.80}$$

Left-interior boundary:

$$c^{CG}_{1/2,\,k-1/2} = \frac{1}{2}(c^{(a,\,0)}_{1/2,\,k-1} + c^{(a,\,0)}_{1/2,\,k}) \tag{5.81}$$

Left-top corner:

$$c^{CG}_{1/2,\,nz+1/2} = \frac{1}{2}(c^{(a,\,0)}_{1/2,\,nz} + c^{(0,\,a)}_{1,\,nz+1/2}) \tag{5.82}$$

Bottom-interior boundary:

$$c^{CG}_{i-1/2,\,1/2} = \frac{1}{2}(c^{(0,\,a)}_{i-1,\,1/2} + c^{(0,\,a)}_{i,\,1/2}) \tag{5.83}$$

Interior nodes:

$$c^{CG}_{i-1/2,\,k-1/2} = \frac{1}{4}(c^{(a,0)}_{i-1/2,\,k} + c^{(a,0)}_{i-1/2,\,k-1} + c^{(0,a)}_{i,\,k-1/2} + c^{(0,a)}_{i-1,\,k-1/2}) \quad \textbf{(5.84)}$$

Top-interior boundary:

$$c^{CG}_{i-1/2,\,nz+1/2} = \frac{1}{2}(c^{(0,a)}_{i-1,\,nz+1/2} + c^{(0,a)}_{i,\,nz+1/2}) \quad \textbf{(5.85)}$$

Right-bottom corner:

$$c^{CG}_{nx+1/2,\,1/2} = \frac{1}{2}(c^{(a,0)}_{nx+1/2,\,1} + c^{(0,a)}_{nx,\,1/2}) \quad \textbf{(5.86)}$$

Right-interior boundary:

$$c^{CG}_{nx+1/2,\,k-1/2} = \frac{1}{2}(c^{(a,0)}_{nx+1/2,\,k-1} + c^{(a,0)}_{nx+1/2,\,k}) \quad \textbf{(5.87)}$$

Right-top corner:

$$c^{CG}_{nx+1/2,\,nz+1/2} = \frac{1}{2}(c^{(a,0)}_{nx+1/2,\,nz} + c^{(0,a)}_{nx,\,nz+1/2}) \quad \textbf{(5.88)}$$

Again *nx* is the number of elements in the *x* direction, *nz* is the number of elements in the *z* direction, and the superscript notation is as in §5.3.1. There is a choice between using the upwind or the average value at the DG interfaces. However, since the adjacent interfaces are already being averaged in this method, it seems natural to use the average value at the element interfaces as well.

5.4.3 Updating the continuous density field

After the CG salinity field has been established using one of the methods outlined

above, the density field is then updated according to the following simplified equation of state:

$$S_{i,k} = 1028.0(1.0 + S_{\beta}(s_{i,k} - S_{\text{ref}})) - 1000.0 \qquad \textbf{(5.89)}$$

where the reference salinity, S_{ref}, is equal to $35.0\,^{0}/_{00}$ and $S_{\beta} = 0.00076$ is the coefficient of saline contraction [*Cushman-Roisin*, 1994]. A more complicated equation can be used when the density field is also dependent upon potential temperature, however, in this study only salinity dependence is examined.

A flow-diagram showing the transport algorithm solution process within the hydrodynamic framework is shown in Figure 5.7. Essentially, the hydrodynamic velocity field is updated, this field is fed into the transport algorithm to determine the new salinity distribution, which is then fed back into the hydrodynamics as a new density field, and the

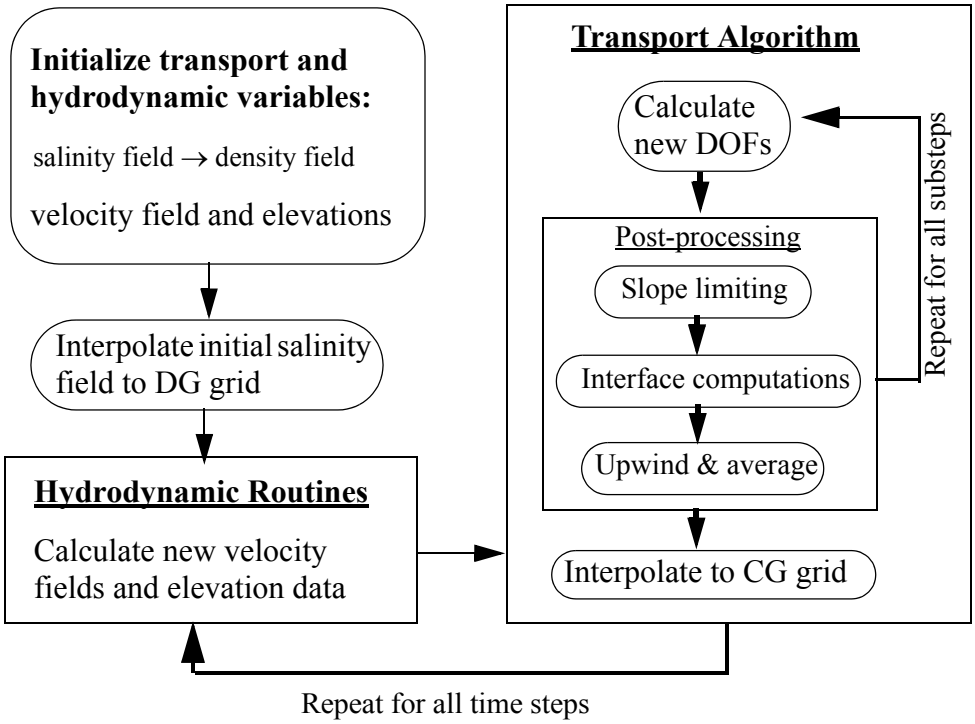


Figure 5.7 Flow diagram of the transport algorithm within the hydrodynamic framework.

next velocity field is established. Thus the most recently calculated hydrodynamic values are used as input in the transport algorithm.

5.5 Interpolation from CG grid space

The LDG discretization of the transport equation described herein was also compared to various CG discretizations for transport as part of another study [*Atkinson, et. al,* 2004]. Details of this study will not be given here, however the initialization of the various algorithms can shed some insight on the LDG discretization itself and will be discussed briefly herein.

In order to ensure a direct comparison between the various discretizations, it was necessary to have the same initial condition for the density field. This required that the initial salinity profile be defined in the CG grid space and then interpolated onto the DG grid. From the separation of variables approach that was taken for the LDG discretization in §5.3, the DG solution is approximated as

$$c(x, z, t) = \hat{c}(z, t)\varphi(x) = c(t)\varphi(x)\psi(z). \qquad \textbf{(5.90)}$$

When the basis functions are substituted for an element, this approximation becomes

$$\begin{aligned} c(x, z, t) &= \hat{c}_{i,k}^{(0,m)}\varphi_0 + \hat{c}_{i,k}^{(1,m)}\varphi_1 \\ &= (c_{i,k}^{(0,0)}\psi_0 + c_{i,k}^{(0,1)}\psi_1)\varphi_0 + (c_{i,k}^{(1,0)}\psi_0 + c_{i,k}^{(1,1)}\psi_1)\varphi_1 \\ &= c_{i,k}^{0,0} + c_{i,k}^{0,1}\psi_1 + c_{i,k}^{(1,0)}\varphi_1 + c_{i,k}^{(1,1)}\psi_1\varphi_1 \end{aligned} \qquad \textbf{(5.91)}$$

where $\varphi_0 = 1$, $\varphi_1 = \xi$, $\xi(x_{i\pm 1/2}) = \pm 1$ and $\psi_0 = 1$, $\psi_1 = \xi$, $\xi(z_{k\pm 1/2}) = \pm 1$ in the dimensionless coordinates introduced earlier. Using these definitions along with Equation (5.91), an expression can be written for each of the CG corner nodes on the element (i,k) in terms of the four DG degrees of freedom:

$$c_{i-1/2,\,k-1/2} = c_{i,k}^{(0,0)} - c_{i,k}^{(0,1)} - (c_{i,k}^{(1,0)} - c_{i,k}^{(1,1)}) \qquad \textbf{(4.92a)}$$

$$c_{i-1/2,\,k+1/2} = c_{i,k}^{(0,0)} + c_{i,k}^{(0,1)} - (c_{i,k}^{(1,0)} + c_{i,k}^{(1,1)}) \qquad \textbf{(4.92b)}$$

$$c_{i+1/2,\,k-1/2} = c_{i,k}^{(0,0)} - c_{i,k}^{(0,1)} + (c_{i,k}^{(1,0)} - c_{i,k}^{(1,1)}) \qquad \textbf{(4.92c)}$$

$$c_{i+1/2,\,k+1/2} = c_{i,k}^{(0,0)} + c_{i,k}^{(0,1)} + (c_{i,k}^{(1,0)} + c_{i,k}^{(1,1)}) \qquad \textbf{(4.92d)}$$

where the grid notation is as in Figure 5.1 on page 233. These equations can be combined in such a way that all of the other variables are eliminated; the resulting relationships provide a formula for each of the DG degrees of freedom in terms of the CG corner nodes.

Take (4.92a) + (4.92b) + (4.92c) + (4.92d) to get

$$c_{i-1/2,\,k-1/2} + c_{i-1/2,\,k+1/2} + c_{i+1/2,\,k-1/2} + c_{i+1/2,\,k+1/2} = 4c_{i,k}^{(0,0)} \,. \qquad \textbf{(5.93)}$$

Take –(4.92a) – (4.92b) + (4.92c) + (4.92d) to get

$$-c_{i-1/2,\,k-1/2} - c_{i-1/2,\,k+1/2} + c_{i+1/2,\,k-1/2} + c_{i+1/2,\,k+1/2} = 4c_{i,k}^{(1,0)} \,. \qquad \textbf{(5.94)}$$

Take –(4.92a) + (4.92b) – (4.92c) + (4.92d) to get

$$-c_{i-1/2,\,k-1/2} + c_{i-1/2,\,k+1/2} - c_{i+1/2,\,k-1/2} + c_{i+1/2,\,k+1/2} = 4c_{i,k}^{(0,1)} \,. \qquad \textbf{(5.95)}$$

Take (4.92a) – (4.92b) – (4.92c) + (4.92d) to get

$$c_{i-1/2,\,k-1/2} - c_{i-1/2,\,k+1/2} - c_{i+1/2,\,k-1/2} + c_{i+1/2,\,k+1/2} = 4c_{i,k}^{(1,1)} \,. \qquad \textbf{(5.96)}$$

Equations (5.93) through (5.96) can then be used to transform a CG data field to the DG grid space.

A better understanding of the DG degrees of freedom can be gleaned from the four previous equations. The element average for the DG solution is simply the average of the

CG corner nodes, which is easily seen from Equation (5.93). The x slope for an element is the average of the x slopes calculated from the CG nodal values along the top and bottom interfaces of the element. This can be seen more clearly if Equation (5.94) is rewritten as

$$c_{i,k}^{(1,0)} = \frac{1}{2}\left[\frac{1}{2}(c_{i+1/2,k-1/2} - c_{i-1/2,k-1/2}) + \frac{1}{2}(c_{i+1/2,k+1/2} - c_{i-1/2,k+1/2})\right] \quad \textbf{(5.97)}$$

where the factors of 1/2 within the square brackets come from the coordinate transformation – the length (and depth) of each dimensionless element is equal to 2.0 – and transform the differences of the CG nodal values to a slope in the x direction. Similarly, the z slope for an element is the average of the z slopes calculated from the CG nodal values along the left and right interfaces of the element, which is clear when Equation (5.95) is rewritten as

$$c_{i,k}^{(0,1)} = \frac{1}{2}\left[\frac{1}{2}(c_{i-1/2,k+1/2} - c_{i-1/2,k-1/2}) + \frac{1}{2}(c_{i+1/2,k+1/2} - c_{i+1/2,k-1/2})\right]. \quad \textbf{(5.98)}$$

The element average, x slope and z slope could also be derived intuitively from a geometric perspective. However, it is not clear from the outset what the interpretation of the x-z slope would be geometrically, and the algebraic definition given in Equation (5.96) can be helpful here. If the formula for the x-z slope is rewritten as

$$c_{i,k}^{(1,1)} = \frac{1}{2}\left[\frac{1}{2}(c_{i+1/2,k+1/2} - c_{i-1/2,k+1/2}) - \frac{1}{2}(c_{i+1/2,k-1/2} - c_{i-1/2,k-1/2})\right] \quad \textbf{(5.99)}$$

then the proper geometric interpretation is that the x-z slope for an element (i,k) is the z slope of the x slopes along the top and bottom interfaces. Or conversely, it could be thought of as the x slope of the z slopes along the left and right interfaces (if the equation was rewritten differently). In either case, all three factors of 1/2 should be viewed as transformation coefficients, as discussed for the x slope, rather than averages.

5.6 Transport model in an x-σ domain

When the bathymetry is not constant, it is often desirable to map the (x, z) domain to the so-called sigma coordinates (x,σ). Here σ varies between σ = a at the free surface, z = ζ (x, t), and σ = b at the bottom, $z = -h(x)$, where $h(x)$ is the bathymetric depth along the channel and a and b are constants. The bathymetry definitions are shown in Figure 5.8.

The coordinate transformation is

$$\sigma = a + \left(\frac{a-b}{H}\right)(z - \zeta) \tag{5.100}$$

and the inverse mapping is

$$z = \left(\frac{\sigma - a}{a - b}\right) H(x, t) + \zeta(x, t) \tag{5.101}$$

where $H(x,t) = h(x) + \zeta(x, t)$ is the total depth. This implies that the z coordinate is a function of σ, x and t. Thus the x and t partial derivatives of z, as given in Equation (5.101) are

$$\frac{\partial z}{\partial x} = \frac{\partial}{\partial x}\left[\left(\frac{\sigma - a}{a - b}\right)(h(x) + \zeta(x, t)) + \zeta(x, t)\right]$$

$$= \left(\frac{\sigma - a}{a - b}\right)\frac{\partial h}{\partial x} + \left(\frac{\sigma - a}{a - b}\right)\frac{\partial \zeta}{\partial x} + \frac{\partial \zeta}{\partial x}$$

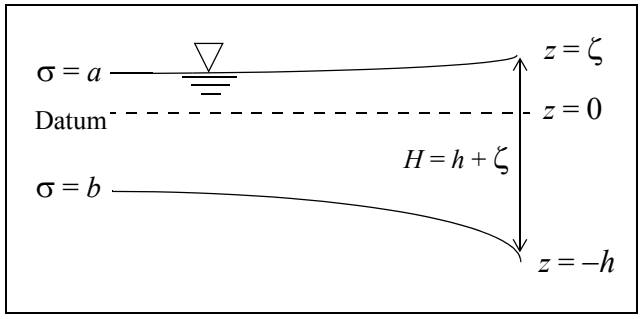


Figure 5.8 Definition of bathymetry terms.

$$= \left(\frac{\sigma - a}{a - b}\right)\frac{\partial h}{\partial x} + \left[\left(\frac{\sigma - a}{a - b}\right) + \left(\frac{a - b}{a - b}\right)\right]\frac{\partial \zeta}{\partial x}$$

$$= \left(\frac{\sigma - a}{a - b}\right)\frac{\partial h}{\partial x} + \left(\frac{\sigma - b}{a - b}\right)\frac{\partial \zeta}{\partial x} \qquad \textbf{(5.102)}$$

and

$$\frac{\partial z}{\partial t} = \frac{\partial}{\partial t}\left[\left(\frac{\sigma - a}{a - b}\right)(h(x) + \zeta(x, t)) + \zeta(x, t)\right]$$

$$= \left[\left(\frac{\sigma - a}{a - b}\right) + \left(\frac{a - b}{a - b}\right)\right]\frac{\partial \zeta}{\partial t}$$

$$= \left(\frac{\sigma - b}{a - b}\right)\frac{\partial \zeta}{\partial t} \ . \qquad \textbf{(5.103)}$$

For any function, f, the derivative in the z direction makes use of the chain rule as follows:

$$\frac{\partial f}{\partial z} = \frac{\partial f}{\partial \sigma}\frac{\partial \sigma}{\partial z} = \frac{\partial}{\partial z}\left[a + \left(\frac{a - b}{H}\right)(z - \zeta)\right]\frac{\partial f}{\partial \sigma}$$

$$= \left(\frac{a - b}{H}\right)\frac{\partial f}{\partial \sigma} \ . \qquad \textbf{(5.104)}$$

Then the derivatives in Equations (5.102) to (5.104) are used to transform the terms of the transport equation, given here again for convenience:

$$\frac{\partial c}{\partial t} + \frac{\partial}{\partial x}\left[uc - N_x\frac{\partial c}{\partial x}\right] + \frac{\partial}{\partial z}\left[wc - N_z\frac{\partial c}{\partial z}\right] = 0 \ . \qquad \textbf{(5.105)}$$

The effect of this transformation is that gradients in z space may be expressed as a gradient in σ space with an additional term to express how z changes as σ changes. Noting that $f = f(x, z, \sigma, t)$ and $\sigma = \sigma(x, z, t)$, the chain rule can be employed to express gradients in x and t as follows:

$$\left.\frac{\partial f}{\partial t}\right|_{\sigma} = \left.\frac{\partial f}{\partial t}\right|_{z} + \frac{\partial f}{\partial z}\frac{\partial z}{\partial t} = \left.\frac{\partial f}{\partial t}\right|_{z} + \frac{\partial f}{\partial \sigma}\frac{\partial \sigma}{\partial z}\frac{\partial z}{\partial t} \tag{5.106}$$

and

$$\left.\frac{\partial f}{\partial x}\right|_{\sigma} = \left.\frac{\partial f}{\partial x}\right|_{z} + \frac{\partial f}{\partial z}\frac{\partial z}{\partial x} = \left.\frac{\partial f}{\partial x}\right|_{z} + \frac{\partial f}{\partial \sigma}\frac{\partial \sigma}{\partial z}\frac{\partial z}{\partial x} \ . \tag{5.107}$$

Since Equation (5.105) is written in terms of gradients in z space, we transform each of the terms according to the previous formulas. The temporal term becomes

$$\left.\frac{\partial c}{\partial t}\right|_{z} = \left.\frac{\partial c}{\partial t}\right|_{\sigma} - \left(\frac{\sigma - a}{a - b}\right)\frac{\partial \zeta}{\partial t}\left(\frac{a - b}{H}\right)\frac{\partial c}{\partial \sigma} \ . \tag{5.108}$$

The x advection term becomes

$$\left.\frac{\partial uc}{\partial x}\right|_{z} = \left.\frac{\partial uc}{\partial x}\right|_{\sigma} - \left[\left(\frac{\sigma - a}{a - b}\right)\frac{\partial h}{\partial x} + \left(\frac{\sigma - b}{a - b}\right)\frac{\partial \zeta}{\partial x}\right]\left(\frac{a - b}{H}\right)\frac{\partial uc}{\partial \sigma} \tag{5.109}$$

and the z advection term becomes

$$\left.\frac{\partial wc}{\partial z}\right|_{z} = \left(\frac{a - b}{H}\right)\frac{\partial wc}{\partial \sigma} \ . \tag{5.110}$$

The z diffusion term becomes

$$\left.\frac{\partial}{\partial z}\left(N_z\frac{\partial c}{\partial z}\right)\right|_{z} = \left(\frac{a - b}{H}\right)^2\frac{\partial}{\partial \sigma}\left(N_z\frac{\partial c}{\partial \sigma}\right) \ . \tag{5.111}$$

The same type of transformation could also be done to the x diffusion term, but the expression is quite cumbersome. Instead, the diffusion coefficient, N_x, is replaced with an effective diffusion coefficient, $N_{x\sigma}$. This is justified by considering that the value of the diffusion parameterization is not known with confidence and is empirically derived; due to this uncertainty, the $N_{x\sigma}$ value may be changed to account for the effect of leaving the

derivatives in their z space form. Thus, the x diffusion in σ coordinates is simply

$$\frac{\partial}{\partial x}\left(N_x \frac{\partial c}{\partial x}\right)\bigg|_z = \frac{\partial}{\partial x_\sigma}\left(N_{x\sigma} \frac{\partial c}{\partial x_\sigma}\right) . \tag{5.112}$$

Therefore, the transformed transport equation is given as

$$\frac{\partial c}{\partial t_\sigma} + \frac{\partial}{\partial x_\sigma}(uc) + \left(\frac{a-b}{H}\right)\frac{\partial}{\partial \sigma}(wc) - \frac{\partial}{\partial x_\sigma}\left(N_{x\sigma}\frac{\partial c}{\partial x_\sigma}\right) -$$

$$\left(\frac{a-b}{H}\right)^2 \frac{\partial}{\partial \sigma}\left(N_z \frac{\partial c}{\partial \sigma}\right) - \Gamma_1 - \Gamma_2 = 0 \tag{5.113}$$

where Γ_1 and Γ_2 are curvature terms, which arise from the coordinate transformation on the temporal and x direction terms. The Γ_1 term comes from the transformation of the temporal term and is calculated as

$$\Gamma_1 = \left(\frac{\sigma - a}{a-b}\right)\frac{\partial \zeta}{\partial t}\left(\frac{a-b}{H}\right)\frac{\partial c}{\partial \sigma} \tag{5.114}$$

while the Γ_2 term comes from the transformation of the x advection term and is calculated as

$$\Gamma_2 = \left[\left(\frac{\sigma - a}{a-b}\right)\frac{\partial h}{\partial x} + \left(\frac{\sigma - b}{a-b}\right)\frac{\partial \zeta}{\partial x}\right]\left(\frac{a-b}{H}\right)\frac{\partial uc}{\partial \sigma} . \tag{5.115}$$

The two curvature terms are combined into a single term, $\Gamma = -\Gamma_1 - \Gamma_2$, and evaluated explicitly. In this way, Γ becomes a known, scalar function.

5.6.1 Finite difference approximations for the curvature terms

As in the ADCIRC hydrodynamics, the various derivatives within the curvature correction terms, Γ_1 and Γ_2, are approximated with simple finite differences. For interior locations, a centered finite difference is used, while one-sided differences are used for the

boundary locations. Meanwhile, a forward Euler scheme is used for the temporal derivative. Where possible, the most recent information is used, since each hydrodynamic and transport variable is calculated in sequence rather than simultaneously. Using these approximations, the derivatives in Γ_1 are given as follows:

$$\frac{\partial \zeta}{\partial t} = \frac{\zeta_{i,k}^{n+1} - \zeta_{i,k}^{n}}{\Delta t} \text{ and} \tag{5.116}$$

$$\frac{\partial c}{\partial \sigma} = \begin{cases} \dfrac{c_{i,k+1}^{n} - c_{i,k}^{n}}{\Delta \sigma_k} & k = 1 \\ \dfrac{c_{i,k}^{n} - c_{i,k-1}^{n}}{\Delta \sigma_{k-1}} & k = nz \\ \dfrac{c_{i,k+1}^{n} - c_{i,k-1}^{n}}{\Delta \sigma_{k-1} + \Delta \sigma_k} & \text{otherwise} \end{cases} . \tag{5.117}$$

Meanwhile, the derivatives in the Γ_2 term are approximated by the following:

$$\frac{\partial h}{\partial x} = \begin{cases} \dfrac{h_{i+1} - h_i}{\Delta x_i} & i = 1 \\ \dfrac{h_i - h_{i-1}}{\Delta x_{i-1}} & i = nx \\ \dfrac{h_{i+1} - h_{i-1}}{\Delta x_{i-1} + \Delta x_i} & \text{otherwise} \end{cases} , \tag{5.118}$$

which is independent of z and t since the bathymetry remains fixed in time and varies only with the horizontal position. Furthermore, the deviation of the water surface is also independent of the vertical position, varying only with time and the horizontal position, so that its derivative is given as

$$\frac{\partial \zeta}{\partial x} = \begin{cases} \dfrac{\zeta_{i+1}^{n+1} - \zeta_i^{n+1}}{\Delta x_i} & i = 1 \\ \dfrac{\zeta_i^{n+1} - \zeta_{i-1}^{n+1}}{\Delta x_{i-1}} & i = nx \\ \dfrac{\zeta_{i+1}^{n+1} - \zeta_{i-1}^{n+1}}{\Delta x_{i-1} + \Delta x_i} & \text{otherwise} \end{cases}, \tag{5.119}$$

where the most recent elevation data is used. Finally, the derivative of the transport advection terms, $u\,c$, are based on the continuous grid space using the most recent velocity calculations, such that the approximations are given as

$$\frac{\partial uc}{\partial \sigma} = \begin{cases} \dfrac{u_{i,k+1}^{n+1} c_{i,k+1}^{n} - u_{i,k}^{n+1} c_{i,k}^{n}}{\Delta \sigma_k} & k = 1 \\ \dfrac{u_{i,k}^{n+1} c_{i,k}^{n} - u_{i,k-1}^{n+1} c_{i,k-1}^{n}}{\Delta \sigma_{k-1}} & k = nz \\ \dfrac{u_{i,k+1}^{n+1} c_{i,k+1}^{n} - u_{i,k-1}^{n+1} c_{i,k-1}^{n}}{\Delta \sigma_{k-1} + \Delta \sigma_k} & \text{otherwise} \end{cases}. \tag{5.120}$$

5.6.2 LDG discretization of x-σ transport

Discretization of Equation (5.113) using the LDG method follows the same procedure as was used in §5.3 for Equation (5.1). The only changes are the addition of the multiplying factor (or stretching factor) on the z advection and diffusion terms, a change from Δz to $\Delta\sigma$, and the addition of the curvature correction terms, Γ. Therefore, the discretization will not be repeated here; instead the alterations will be described in words and the final equations given.

As given in Equation (5.113), a constant stretching factor equal to $(a–b) / H_i$, where the i index indicates the i^{th} element in the x direction, will multiply each discrete z advection

term and the square of this factor will multiply each discrete z diffusion term. Also, H_i is defined to be the average value for the element since linear approximating functions are used for $\zeta(x)$. This factor is constant with depth and varies only according to the x position, as the total depth changes along the channel length. Also, the differentials must be changed to the σ coordinate system, such that $\Delta\sigma_k = \sigma_{k+1} - \sigma_k$, as was the case for Δz.

The final step, to complete the transformation to σ coordinates, is to discretize the curvature correction terms, Γ. Since these terms are dependent upon the bathymetry, sea surface elevation, horizontal velocity (u), and the concentration (c), a choice must be made as to whether they should be continuous or discontinuous variables. Concentration is the only dependent variable that is assumed to be discontinuous, so it would be difficult to define the entire correction term as discontinuous. Additionally, the DG concentration values have already been interpolated to the CG grid in order to update the density field. Therefore, it seems most natural to treat the curvature correction terms, Γ, as continuous variables.

With this assumption, the curvature correction terms are approximated in the x direction with continuous linear Lagrange functions giving

$$\hat{\Gamma}_{i,k} = \hat{\Gamma}_{i-1/2,k}\phi_0 + \hat{\Gamma}_{i+1/2,k}\phi_1 \qquad \textbf{(5.121)}$$

where the hat notation indicates that the coefficients are still functions of z. The weighted residual over an element, with discontinuous weight function φ, then becomes

$$((\hat{\Gamma}_{i-1/2,k}\phi_0 + \hat{\Gamma}_{i+1/2,k}\phi_1),\varphi_j)_{\Omega_e} \qquad \textbf{(5.122)}$$

where the continuous and discontinuous bases, are as in §5.3.1. This results in two equations for the j degrees of freedom 0 and 1:

$$\frac{\Delta x_i}{2}(\hat{\Gamma}_{i-1/2,k} + \hat{\Gamma}_{i+1/2,k}) \text{ for } j = 0 \qquad \textbf{(5.123)}$$

and

$$\frac{\Delta x_i}{2}\frac{1}{3}(-\hat{\Gamma}_{i-1/2,k}+\hat{\Gamma}_{i+1/2,k}) \text{ for } j=1. \tag{5.124}$$

Now the j equations are approximated in the σ direction with linear Lagrange functions such that

$$\Gamma_{i,k} = \Gamma_{i,k-1/2}\zeta_0+\Gamma_{i,k+1/2}\zeta_1 \tag{5.125}$$

where the hat notation has been removed to indicate that the coefficients are functions of time alone. After making this substitution, each of the j equations are integrated in the σ direction. For j=0 this gives

$$\frac{\Delta x_i}{2}[((\Gamma_{i-1/2,k-1/2}\zeta_0+\Gamma_{i-1/2,k+1/2}\zeta_1)\psi_m)_{\Omega_e}+$$
$$((\Gamma_{i+1/2,k-1/2}\zeta_0+\Gamma_{i+1/2,k+1/2}\zeta_1)\psi_m)_{\Omega_e}] \tag{5.126}$$

where the continuous bases and discontinuous weighting functions are as in §5.3.2 and the integrations result in two equations:

<u>(j=0, m=0)</u>

$$\frac{\Delta x_i}{2}\frac{\Delta\sigma_k}{2}(\Gamma_{i-1/2,k-1/2}+\Gamma_{i-1/2,k+1/2}+\Gamma_{i+1/2,k-1/2}+\Gamma_{i+1/2,k+1/2}) \tag{5.127}$$

<u>(j=0, m=1)</u>

$$\frac{\Delta x_i}{2}\frac{\Delta\sigma_k}{6}(-\Gamma_{i-1/2,k-1/2}+\Gamma_{i-1/2,k+1/2}-\Gamma_{i+1/2,k-1/2}+\Gamma_{i+1/2,k+1/2}) \tag{5.128}$$

For j=1 this gives

$$\frac{\Delta x_i}{6}[(-(\Gamma_{i-1/2,\,k-1/2}\zeta_0 + \Gamma_{i-1/2,\,k+1/2}\zeta_1)\psi_m)_{\Omega_e} +$$

$$((\Gamma_{i+1/2,\,k-1/2}\zeta_0 + \Gamma_{i+1/2,\,k+1/2}\zeta_1)\psi_m)_{\Omega_e}] \qquad \textbf{(5.129)}$$

where again the continuous bases and discontinuous weighting functions are as in §5.3.2 and the integrations result in two equations:

(j=1, m=0)

$$\frac{\Delta x_i}{6}\frac{\Delta\sigma_k}{2}(-\Gamma_{i-1/2,\,k-1/2} - \Gamma_{i-1/2,\,k+1/2} + \Gamma_{i+1/2,\,k-1/2} + \Gamma_{i+1/2,\,k+1/2}) \qquad \textbf{(5.130)}$$

(j=1, m=1)

$$\frac{\Delta x_i}{6}\frac{\Delta\sigma_k}{6}(\Gamma_{i-1/2,\,k-1/2} - \Gamma_{i-1/2,\,k+1/2} - \Gamma_{i+1/2,\,k-1/2} + \Gamma_{i+1/2,\,k+1/2}). \qquad \textbf{(5.131)}$$

Equations (5.127), (5.128), (5.130) and (5.131), respectively, are added to the left-hand side of Equations (5.34), (5.37), (5.40), and (5.43) in Table 5.2. These curvature correction terms are evaluated using the approximations given above in §5.6.1.

This completes the transformation from z coordinates to σ coordinates. The boundary conditions and post-processing from the previous derivation in §5.3.3 and §5.3.4 are still valid for these transformed equations. The interpolations to the CG grid space described in §5.4 remain the same, except that Δz is replaced by $\Delta\sigma$. The discrete equations for an interior element in (x,σ) space are given below in Table 5.3.

Table 5.3 Spatial discretization of the x-σ transport equation for an interior element (using linear approximating spaces).

(j = 0, m = 0)
$\Delta x_i\Delta\sigma_k\frac{d}{dt}c_{i,k}^{(0,0)} + \frac{\Delta\sigma_k}{2}[2\gamma_{i+1/2,\,k}^{(a,0)} - 2\gamma_{i-1/2,\,k}^{(a,0)}] + \frac{\Delta x_i}{2}\left(\frac{a-b}{H_i}\right)^2[2\chi_{i,\,k+1/2}^{(0,a)} - 2\chi_{i,\,k-1/2}^{(0,a)}] +$

Table 5.3 Spatial discretization of the x-σ transport equation for an interior element (using linear approximating spaces).

$$\frac{\Delta\sigma_k}{2}\left[u_{i+1/2,k-1/2}\left(c^{(u,0)}_{i+1/2,k}-\frac{1}{3}c^{(u,1)}_{i+1/2,k}\right)+u_{i+1/2,k+1/2}\left(c^{(u,0)}_{i+1/2,k}+\frac{1}{3}c^{(u,1)}_{i+1/2,k}\right)\right]-$$

$$\frac{\Delta\sigma_k}{2}\left[u_{i-1/2,k-1/2}\left(c^{(u,0)}_{i-1/2,k}-\frac{1}{3}c^{(u,1)}_{i-1/2,k}\right)+u_{i-1/2,k+1/2}\left(c^{(u,0)}_{i-1/2,k}+\frac{1}{3}c^{(u,1)}_{i-1/2,k}\right)\right]+$$

$$\frac{\Delta x_i}{2}\left(\frac{a-b}{H_i}\right)\left[w_{i-1/2,k+1/2}\left(c^{(0,u)}_{i,k+1/2}-\frac{1}{3}c^{(1,u)}_{i,k+1/2}\right)\right]+$$

$$\frac{\Delta x_i}{2}\left(\frac{a-b}{H_i}\right)\left[w_{i+1/2,k+1/2}\left(c^{(0,u)}_{i,k+1/2}+\frac{1}{3}c^{(1,u)}_{i,k+1/2}\right)\right]-$$

$$\frac{\Delta x_i}{2}\left(\frac{a-b}{H_i}\right)\left[w_{i-1/2,k-1/2}\left(c^{(0,u)}_{i,k-1/2}-\frac{1}{3}c^{(1,u)}_{i,k-1/2}\right)\right]-$$

$$\frac{\Delta x_i}{2}\left(\frac{a-b}{H_i}\right)\left[w_{i+1/2,k-1/2}\left(c^{(0,u)}_{i,k-1/2}+\frac{1}{3}c^{(1,u)}_{i,k-1/2}\right)\right]+$$

$$\frac{\Delta x_i}{2}\frac{\Delta\sigma_k}{2}[\Gamma_{i-1/2,k-1/2}+\Gamma_{i-1/2,k+1/2}+\Gamma_{i+1/2,k-1/2}+\Gamma_{i+1/2,k+1/2}]=0 \qquad \textbf{(5.132)}$$

$$\frac{\Delta x_i}{N_{x\sigma}}\gamma^{(0,0)}_{i,k}+c^{(a,0)}_{i+1/2,k}-c^{(a,0)}_{i-1/2,k}=0 \qquad \textbf{(5.133)}$$

$$\frac{\Delta\sigma_k}{N_z}\chi^{(0,0)}_{i,k}+c^{(0,a)}_{i,k+1/2}-c^{(0,a)}_{i,k-1/2}=0 \qquad \textbf{(5.134)}$$

($j=0, m=1$)

$$\Delta x_i\frac{\Delta\sigma_k}{3}\frac{d}{dt}c^{(0,1)}_{i,k}+\frac{\Delta\sigma_k}{2}\left[\frac{2}{3}\gamma^{(a,1)}_{i+1/2,k}+\frac{2}{3}\gamma^{(a,1)}_{i-1/2,k}\right]+$$

$$\frac{\Delta x_i}{2}\left(\frac{a-b}{H_i}\right)^2[2\chi^{(0,a)}_{i,k+1/2}+2\chi^{(0,a)}_{i,k-1/2}-4\chi^{(0,0)}_{i,k}]+$$

$$\frac{\Delta\sigma_k}{6}[u_{i+1/2,k-1/2}(-c^{(u,0)}_{i+1/2,k}+c^{(u,1)}_{i+1/2,k})+u_{i+1/2,k+1/2}(c^{(u,0)}_{i+1/2,k}+c^{(u,1)}_{i+1/2,k})]-$$

$$\frac{\Delta\sigma_k}{6}[u_{i-1/2,k-1/2}(-c^{(u,0)}_{i-1/2,k}+c^{(u,1)}_{i-1/2,k})+u_{i-1/2,k+1/2}(c^{(u,0)}_{i-1/2,k}+c^{(u,1)}_{i-1/2,k})]-$$

Table 5.3 Spatial discretization of the x-σ transport equation for an interior element (using linear approximating spaces).

$$\frac{\Delta x_i}{2}\left(\frac{a-b}{H_i}\right)\left[w_{i-1/2,\,k-1/2}\left(c_{i,k}^{(0,0)}-\frac{1}{3}c_{i,k}^{(0,1)}-\frac{1}{3}c_{i,k}^{(1,0)}+\frac{1}{9}c_{i,k}^{(1,1)}\right)\right]-$$

$$\frac{\Delta x_i}{2}\left(\frac{a-b}{H_i}\right)\left[w_{i-1/2,\,k+1/2}\left(c_{i,k}^{(0,0)}+\frac{1}{3}c_{i,k}^{(0,1)}-\frac{1}{3}c_{i,k}^{(1,0)}-\frac{1}{9}c_{i,k}^{(1,1)}\right)\right]-$$

$$\frac{\Delta x_i}{2}\left(\frac{a-b}{H_i}\right)\left[w_{i+1/2,\,k-1/2}\left(c_{i,k}^{(0,0)}-\frac{1}{3}c_{i,k}^{(0,1)}+\frac{1}{3}c_{i,k}^{(1,0)}-\frac{1}{9}c_{i,k}^{(1,1)}\right)\right]-$$

$$\frac{\Delta x_i}{2}\left(\frac{a-b}{H_i}\right)\left[w_{i+1/2,\,k+1/2}\left(c_{i,k}^{(0,0)}+\frac{1}{3}c_{i,k}^{(0,1)}+\frac{1}{3}c_{i,k}^{(1,0)}+\frac{1}{9}c_{i,k}^{(1,1)}\right)\right]+$$

$$\frac{\Delta x_i}{2}\left(\frac{a-b}{H_i}\right)\left[w_{i-1/2,\,k+1/2}\left(c_{i,k+1/2}^{(0,u)}-\frac{1}{3}c_{i,k+1/2}^{(1,u)}\right)\right]+$$

$$\frac{\Delta x_i}{2}\left(\frac{a-b}{H_i}\right)\left[w_{i+1/2,\,k+1/2}\left(c_{i,k+1/2}^{(0,u)}+\frac{1}{3}c_{i,k+1/2}^{(1,u)}\right)\right]+$$

$$\frac{\Delta x_i}{2}\left(\frac{a-b}{H_i}\right)\left[w_{i-1/2,\,k-1/2}\left(c_{i,k-1/2}^{(0,u)}-\frac{1}{3}c_{i,k-1/2}^{(1,u)}\right)\right]+$$

$$\frac{\Delta x_i}{2}\left(\frac{a-b}{H_i}\right)\left[w_{i+1/2,\,k-1/2}\left(c_{i,k-1/2}^{(0,u)}+\frac{1}{3}c_{i,k-1/2}^{(1,u)}\right)\right]+$$

$$\frac{\Delta x_i}{6}\frac{\Delta\sigma_k}{2}\left[-\Gamma_{i-1/2,\,k-1/2}+\Gamma_{i-1/2,\,k+1/2}-\Gamma_{i+1/2,\,k-1/2}+\Gamma_{i+1/2,\,k+1/2}\right]=0 \qquad \textbf{(5.135)}$$

$$\frac{\Delta x_i}{N_{x\sigma}}\gamma_{i,k}^{(0,1)}+c_{i+1/2,\,k}^{(a,1)}-c_{i-1/2,\,k}^{(a,1)}=0 \qquad \textbf{(5.136)}$$

$$\frac{\Delta\sigma_k}{3N_z}\chi_{i,k}^{(0,1)}-2c_{i,k}^{(0,0)}+c_{i,k+1/2}^{(0,a)}+c_{i,k-1/2}^{(0,a)}=0 \qquad \textbf{(5.137)}$$

($j=1,\ m=0$)

$$\frac{\Delta x_i}{3}\Delta\sigma_k\frac{d}{dt}c_{i,k}^{(1,0)}+\frac{\Delta\sigma_k}{2}\left[2\gamma_{i+1/2,\,k}^{(a,0)}+2\gamma_{i-1/2,\,k}^{(a,0)}-4\gamma_{i,k}^{(0,0)}\right]+$$

$$\frac{\Delta x_i}{2}\left(\frac{a-b}{H_i}\right)^2\left[\frac{2}{3}\chi_{i,k+1/2}^{(1,a)}+\frac{2}{3}\chi_{i,k-1/2}^{(1,a)}\right]-$$

Table 5.3 Spatial discretization of the x-σ transport equation for an interior element (using linear approximating spaces).

$$\frac{\Delta\sigma_k}{2}\left[u_{i-1/2,\,k-1/2}\left(c_{i,k}^{(0,0)}-\frac{1}{3}c_{i,k}^{(0,1)}-\frac{1}{3}c_{i,k}^{(1,0)}+\frac{1}{9}c_{i,k}^{(1,1)}\right)\right]-$$

$$\frac{\Delta\sigma_k}{2}\left[u_{i-1/2,\,k+1/2}\left(c_{i,k}^{(0,0)}+\frac{1}{3}c_{i,k}^{(0,1)}-\frac{1}{3}c_{i,k}^{(1,0)}-\frac{1}{9}c_{i,k}^{(1,1)}\right)\right]-$$

$$\frac{\Delta\sigma_k}{2}\left[u_{i+1/2,\,k-1/2}\left(c_{i,k}^{(0,0)}-\frac{1}{3}c_{i,k}^{(0,1)}+\frac{1}{3}c_{i,k}^{(1,0)}-\frac{1}{9}c_{i,k}^{(1,1)}\right)\right]-$$

$$\frac{\Delta\sigma_k}{2}\left[u_{i+1/2,\,k+1/2}\left(c_{i,k}^{(0,0)}+\frac{1}{3}c_{i,k}^{(0,1)}+\frac{1}{3}c_{i,k}^{(1,0)}+\frac{1}{9}c_{i,k}^{(1,1)}\right)\right]+$$

$$\frac{\Delta\sigma_k}{2}\left[u_{i+1/2,\,k-1/2}\left(c_{i+1/2,\,k}^{(u,0)}-\frac{1}{3}c_{i+1/2,\,k}^{(u,1)}\right)+u_{i+1/2,\,k+1/2}\left(c_{i+1/2,\,k}^{(u,0)}+\frac{1}{3}c_{i+1/2,\,k}^{(u,1)}\right)\right]+$$

$$\frac{\Delta\sigma_k}{2}\left[u_{i-1/2,\,k-1/2}\left(c_{i-1/2,\,k}^{(u,0)}-\frac{1}{3}c_{i-1/2,\,k}^{(u,1)}\right)+u_{i-1/2,\,k+1/2}\left(c_{i-1/2,\,k}^{(u,0)}+\frac{1}{3}c_{i-1/2,\,k}^{(u,1)}\right)\right]+$$

$$\frac{\Delta x_i}{6}\left(\frac{a-b}{H_i}\right)[w_{i-1/2,\,k+1/2}(-c_{i,k+1/2}^{(0,u)}+c_{i,k+1/2}^{(1,u)})]+$$

$$\frac{\Delta x_i}{6}\left(\frac{a-b}{H_i}\right)[w_{i+1/2,\,k+1/2}(c_{i,k+1/2}^{(0,u)}+c_{i,k+1/2}^{(1,u)})]-$$

$$\frac{\Delta x_i}{6}\left(\frac{a-b}{H_i}\right)[w_{i-1/2,\,k-1/2}(-c_{i,k-1/2}^{(0,u)}+c_{i,k-1/2}^{(1,u)})]-$$

$$\frac{\Delta x_i}{6}\left(\frac{a-b}{H_i}\right)[w_{i+1/2,\,k-1/2}(c_{i,k-1/2}^{(0,u)}+c_{i,k-1/2}^{(1,u)})]+$$

$$\frac{\Delta x_i}{2}\frac{\Delta\sigma_k}{6}[-\Gamma_{i-1/2,\,k-1/2}-\Gamma_{i-1/2,\,k+1/2}+\Gamma_{i+1/2,\,k-1/2}+\Gamma_{i+1/2,\,k+1/2}]=0 \qquad \textbf{(5.138)}$$

$$\frac{\Delta x_i}{3N_{x\sigma}}\gamma_{i,k}^{(1,0)}-2c_{i,k}^{(0,0)}+c_{i+1/2,\,k}^{(a,0)}+c_{i-1/2,\,k}^{(a,0)}=0 \qquad \textbf{(5.139)}$$

$$\frac{\Delta\sigma_k}{N_z}\chi_{i,k}^{(1,0)}+c_{i,k+1/2}^{(1,a)}-c_{i,k-1/2}^{(1,a)}=0 \qquad \textbf{(5.140)}$$

Table 5.3 Spatial discretization of the x-σ transport equation for an interior element (using linear approximating spaces).

$(j = 1, m = 1)$
$\frac{\Delta x_i}{3}\frac{\Delta \sigma_k}{3}\frac{d}{dt}c_{i,k}^{(1,1)} + \frac{\Delta \sigma_k}{2}\left[\frac{2}{3}\gamma_{i+1/2,k}^{(a,1)} + \frac{2}{3}\gamma_{i-1/2,k}^{(a,1)} - \frac{4}{3}\gamma_{i,k}^{(0,1)}\right] +$
$\frac{\Delta x_i}{2}\left(\frac{a-b}{H_i}\right)^2\left[\frac{2}{3}\chi_{i,k+1/2}^{(1,a)} + \frac{2}{3}\chi_{i,k-1/2}^{(1,a)} - \frac{4}{3}\chi_{i,k}^{(1,0)}\right] -$
$\frac{\Delta \sigma_k}{6}\left[u_{i-1/2,k-1/2}\left(-c_{i,k}^{(0,0)} + c_{i,k}^{(0,1)} + \frac{1}{3}c_{i,k}^{(1,0)} - \frac{1}{3}c_{i,k}^{(1,1)}\right)\right] -$
$\frac{\Delta \sigma_k}{6}\left[u_{i-1/2,k+1/2}\left(c_{i,k}^{(0,0)} + c_{i,k}^{(0,1)} - \frac{1}{3}c_{i,k}^{(1,0)} - \frac{1}{3}c_{i,k}^{(1,1)}\right)\right] -$
$\frac{\Delta \sigma_k}{6}\left[u_{i+1/2,k-1/2}\left(-c_{i,k}^{(0,0)} + c_{i,k}^{(0,1)} - \frac{1}{3}c_{i,k}^{(1,0)} + \frac{1}{3}c_{i,k}^{(1,1)}\right)\right] -$
$\frac{\Delta \sigma_k}{6}\left[u_{i+1/2,k+1/2}\left(c_{i,k}^{(0,0)} + c_{i,k}^{(0,1)} + \frac{1}{3}c_{i,k}^{(1,0)} + \frac{1}{3}c_{i,k}^{(1,1)}\right)\right] +$
$\frac{\Delta \sigma_k}{6}[u_{i+1/2,k-1/2}(-c_{i+1/2,k}^{(u,0)} + c_{i+1/2,k}^{(u,1)}) + u_{i+1/2,k+1/2}(c_{i+1/2,k}^{(u,0)} + c_{i+1/2,k}^{(u,1)})] +$
$\frac{\Delta \sigma_k}{6}[u_{i-1/2,k-1/2}(-c_{i-1/2,k}^{(u,0)} + c_{i-1/2,k}^{(u,1)}) + u_{i-1/2,k+1/2}(c_{i-1/2,k}^{(u,0)} + c_{i-1/2,k}^{(u,1)})] -$
$\frac{\Delta x_i}{6}\left(\frac{a-b}{H_i}\right)\left[w_{i-1/2,k-1/2}\left(-c_{i,k}^{(0,0)} + \frac{1}{3}c_{i,k}^{(0,1)} + c_{i,k}^{(1,0)} - \frac{1}{3}c_{i,k}^{(1,1)}\right)\right] -$
$\frac{\Delta x_i}{6}\left(\frac{a-b}{H_i}\right)\left[w_{i-1/2,k+1/2}\left(-c_{i,k}^{(0,0)} - \frac{1}{3}c_{i,k}^{(0,1)} + c_{i,k}^{(1,0)} + \frac{1}{3}c_{i,k}^{(1,1)}\right)\right] -$
$\frac{\Delta x_i}{6}\left(\frac{a-b}{H_i}\right)\left[w_{i+1/2,k-1/2}\left(c_{i,k}^{(0,0)} - \frac{1}{3}c_{i,k}^{(0,1)} + c_{i,k}^{(1,0)} - \frac{1}{3}c_{i,k}^{(1,1)}\right)\right] -$
$\frac{\Delta x_i}{6}\left(\frac{a-b}{H_i}\right)\left[w_{i+1/2,k+1/2}\left(c_{i,k}^{(0,0)} + \frac{1}{3}c_{i,k}^{(0,1)} + c_{i,k}^{(1,0)} + \frac{1}{3}c_{i,k}^{(1,1)}\right)\right] +$
$\frac{\Delta x_i}{6}\left(\frac{a-b}{H_i}\right)[w_{i-1/2,k+1/2}(-c_{i,k+1/2}^{(0,u)} + c_{i,k+1/2}^{(1,u)})] +$

Table 5.3 Spatial discretization of the x-σ transport equation for an interior element (using linear approximating spaces).

$$\frac{\Delta x_i}{6}\left(\frac{a-b}{H_i}\right)[w_{i+1/2,\,k+1/2}(c^{(0,u)}_{i,\,k+1/2}+c^{(1,u)}_{i,\,k+1/2})]+$$

$$\frac{\Delta x_i}{6}\left(\frac{a-b}{H_i}\right)[w_{i-1/2,\,k-1/2}(-c^{(0,u)}_{i,\,k-1/2}+c^{(1,u)}_{i,\,k-1/2})]+$$

$$\frac{\Delta x_i}{6}\left(\frac{a-b}{H_i}\right)[w_{i+1/2,\,k-1/2}(c^{(0,u)}_{i,\,k-1/2}+c^{(1,u)}_{i,\,k-1/2})]+$$

$$\frac{\Delta x_i}{6}\frac{\Delta\sigma_k}{6}[\Gamma_{i-1/2,\,k-1/2}-\Gamma_{i-1/2,\,k+1/2}-\Gamma_{i+1/2,\,k-1/2}+\Gamma_{i+1/2,\,k+1/2}]\;=\;0 \qquad \textbf{(5.141)}$$

$$\frac{\Delta x_i}{3N_{x\sigma}}\gamma^{(1,1)}_{i,\,k}-2c^{(0,1)}_{i,\,k}+c^{(a,1)}_{i+1/2,\,k}+c^{(a,1)}_{i-1/2,\,k}\;=\;0 \qquad \textbf{(5.142)}$$

$$\frac{\Delta\sigma_k}{3N_z}\chi^{(1,1)}_{i,\,k}-2c^{(1,0)}_{i,\,k}+c^{(1,a)}_{i,\,k+1/2}+c^{(1,a)}_{i,\,k-1/2}\;=\;0 \qquad \textbf{(5.143)}$$

5.7 Select results with the x-z transport model

5.7.1 Verification with 1D analytical solution

The LDG transport algorithm in x-z coordinates, as described above in §5.3, is verified with an analytical solution for breakthrough plumes with advection and diffusion in one spatial direction. A rectangular domain with constant grid spacing in the x and z coordinates is used. For each coordinate space, a 1D problem is set up by setting the velocity and diffusion coefficient for the cross-stream coordinate direction equal to zero. For the coordinate of interest, Type I boundary conditions with $\tilde{c} = 1.0$ are used; Type II conditions with an inflow concentration of 0.0 are used for the cross-stream coordinate.

A domain with horizontal dimension of 1000m and constant depth of 20m was used as the test domain for the grid convergence studies. The domain was discretized with uniform horizontal and vertical grid spacing. For the x coordinate study, the vertical grid spacing was held constant at a value of $\Delta z = 2.0$ m, while the horizontal grid was resolved;

and for the z coordinate study, the horizontal grid spacing was held constant at a value of Δx = 100.0 m, while the vertical grid was resolved. The mid-depth (or mid-channel for the z coordinate study) solution was compared to the exact solution and L_2 and L_∞ error norms were recorded as the grid was refined. The L_2 error is computed over space and time as

$$L_2\ error\ =\ \sqrt{\left(\sum_{t=1}^{nt}\left(\left(\sum_{i=1}^{nx}\sum_{k=1}^{nz}(C(x,z,t)-c_{i,k})^2\right)/(nx\ nz)\right)\right)/nt} \qquad \textbf{(5.144)}$$

and the L_∞ error is taken to be the maximum error over all space and time given as

$$L_\infty\ =\ \sqrt{Max[(C(x,z,t)-c_{i,k})^2]} \qquad \forall(i,k,t) \qquad \textbf{(5.145)}$$

where nt is the number of temporal outputs from the simulation and squared errors are used in the L_∞ norm to eliminate the signs. The analytical solution $C(x$ or $z,\ t)$ for a one-dimensional solute breakthrough, as presented by *Knox et al.* [1993], in the x or z coordinate direction is given by Equation (5.148) or (5.149), respectively. The convergence of the error norms, as the grids are refined, are presented in tabular form and in graphical log-log plots.

Two dimensionless numbers are used to characterize the simulations: the Courant number, given for the x and z-coordinate directions as

$$\mathrm{Cr}_x\ =\ u\Delta t/\Delta x \text{ and } \mathrm{Cr}_z\ =\ w\Delta t/\Delta z\,, \qquad \textbf{(5.146)}$$

and the Peclet number, given for the x and z-coordinate directions as

$$\mathrm{Pe}_x\ =\ u\Delta x/N_x \text{ and } \mathrm{Pe}_z\ =\ w\Delta z/N_z\ . \qquad \textbf{(5.147)}$$

The Courant number is often used as a measure of the temporal stability limit and for these tests it was kept low by choosing Δt = 0.01 seconds for all simulations. Tests were run with smaller Δt values for select diffusivities and grid spacings, but the magnitudes of the error

norms did not change significantly. Temporal convergence is discussed in §5.7.3. Meanwhile, the Peclet number is often used as a measure of whether the transport is advection (high Pe) or diffusion dominated (low Pe).

Verification of *x*-coordinate direction

To set up a test case for the x coordinate, the vertical velocity and diffusion coefficient, w and N_z, are set equal to zero and the horizontal velocity, u, is set equal to unity. The solution along the mid-depth is plotted after a simulated 500 sec. and compared to the analytical solution given by,

$$C(x,t) = \frac{c_o}{2}\left[erfc\frac{(x-ut)}{2\sqrt{N_x t}} + \exp\frac{ux}{N_x}erfc\frac{(x+ut)}{2\sqrt{N_x t}}\right] \quad \textbf{(5.148)}$$

where *erfc* represents the complementary error function, c_o is the strength of the boundary condition, N_x is the diffusivity, x is the position along the channel length, and t is the time. For this study, four diffusivities ranging from advection-dominated flows (N_x = 0.001 m^2/ sec) to diffusion-dominated flows (N_x = 25.0 m^2/sec) are examined. For each diffusivity, grid convergence results are generated from simulation output at 20 second intervals over the length of the 500 second simulation, such that nt = 25 for the L_2 error norm.

The analytical solutions for the study diffusivities N_x= 0.001, 1.0, 10.0, and 25.0 m^2/sec at t = 500 seconds are plotted in Figure 5.9. Notice that as the diffusion coefficient is increased the breakthrough curve is not only more dispersed (wider spread), but also that higher concentrations are progressing further into the channel. Thus, the curves are no longer symmetric as the diffusion increases.

The maximum cross-stream variation for the domain was also monitored for each run and additional runs were made with increased resolution in the cross-stream (z) direction. As expected, the cross-stream variation was zero and no significant change was noticed in the error norms for this increased resolution, indicating that the solution is indeed

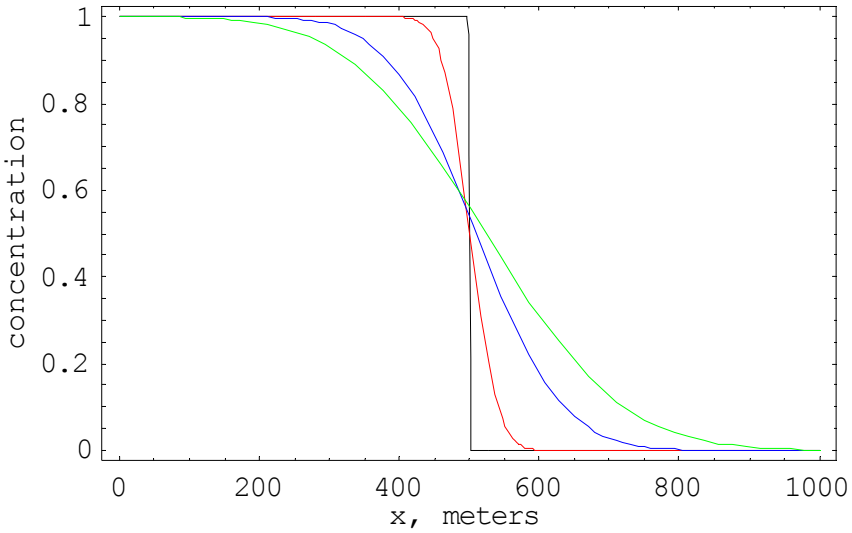


Figure 5.9 Analytical solution for advection-dispersion breakthrough curves in the x coordinate for N_x = 0.001, 1.0, 10.0 and 25.0 m^2/sec at t = 500 seconds.

one-dimensional in the x direction and the errors are dominated by the x grid resolution. For each diffusivity, the simulated results for various grid spacings are compared to the analytical solution (plotted with solid black line), log-log plots of the error norms are presented, and a summary table of the convergence results is given.

$N_x = 0.001\ m^2/sec$

Figures 5.10 and 5.11 show the simulation results and log-log plots of error norms for the advection-dominated test case (Peclet numbers from 2,500 to 200,000). The simulation results at resolutions of Δx = 5m and 10m are not shown as they lie close to the analytical curve and make the graph too crowded. Table 5.4 presents a summary of the error norms.

Notice that the L_∞ errors do not change significantly as the grid is refined. This is due to the sharpness of the front. As the grid is refined, the simulated results lie along the analytical curve at all points graphically (the red curve in Figure 5.10). However, at the

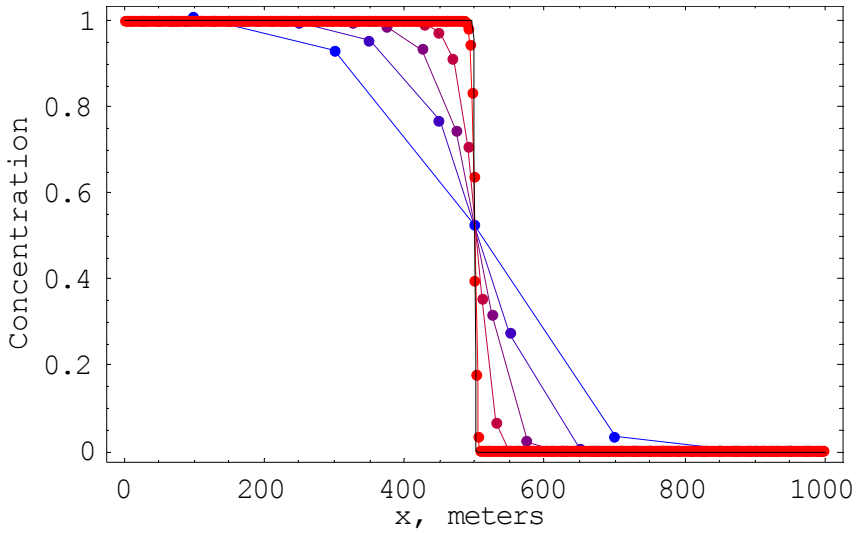


Figure 5.10 Simulation results for advection-dominated test case (N_x = 0.001 m^2/sec) at time t = 500 seconds. Grid resolution, Δx [100m to 2.5m].

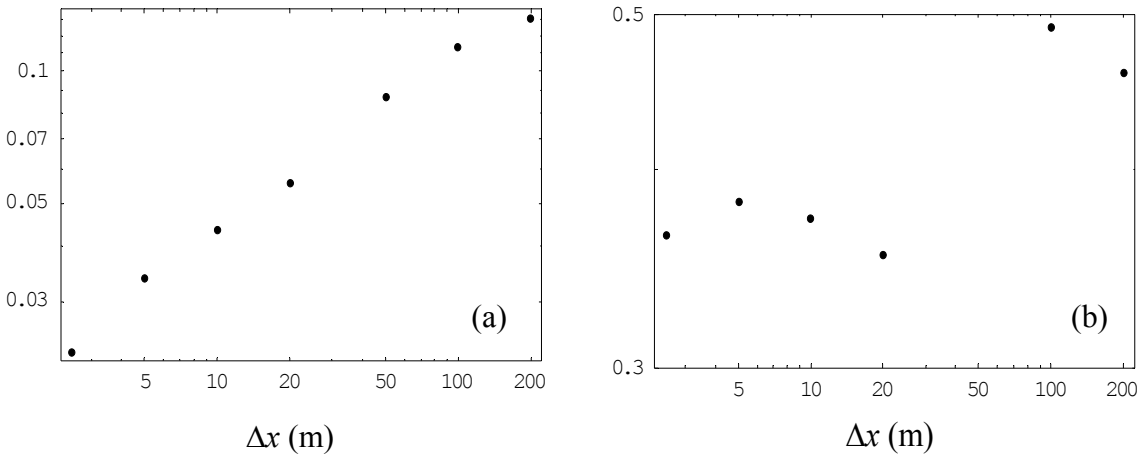


Figure 5.11 Log-log plot of spatial error norms for N_x = 0.001 m^2/sec : (a) L_2 errors and (b) L_{inf} errors.

jump itself, the simulated results have many data points along the jump while the analytical solution is an instantaneous discontinuity. Therefore, the maximum error over the domain will always be located at the jump discontinuity.

Table 5.4 Grid convergence results for $N_x = 0.001$ m^2/sec.

Δx (m)	Pe_x	Cr_x	L_2	L_∞
200	200,000	0.0005	0.131661	0.459661
100	100,000	0.0001	0.112492	0.491039
50	50,000	0.0002	0.087171	0.502629
20	20,000	0.0005	0.055514	0.353272
10	10,000	0.001	0.043693	0.372653
5	5,000	0.002	0.033719	0.381199
2.5	2,500	0.004	0.022933	0.363765
Peak convergence rate:			0.556	0.385
Average convergence rate:			0.394	0.115
Best fit convergence rate:			0.402	0.075

The peak convergence rate is the largest change between successive grid resolutions and theoretically should reach a value of 2.0 for linear interpolating functions in space. However, due to the severity of this problem (a sharp front), this theoretical value was not attained. The average convergence rate is computed as the arithmetic average of the rates between successive grid resolutions, and the best-fit convergence rate is the slope of a best-fit linear regression through the log-log plots.

$N_x = 1.0$ m^2/sec

Figures 5.12 and 5.13 show the simulation results and error norms for a slightly more diffusive test case (Peclet numbers from 2.5 to 200). Table 5.5 presents a summary of the error norms. Notice that less resolution is required to get a more accurate simulation when some diffusion is added. For example, with $N_x = 0.001$ m^2/sec the L_2 error at Δx = 2.5m is 0.0229, but for $N_x = 1.0$ m^2/sec the L_2 error at Δx = 10m is 0.0035 – an order of magnitude improvement in error with one-fourth the number of elements. The errors are converging to zero and the convergence rates approach the theoretical value of 2.0 for linear interpolants as the grid is refined further, and the peak convergence rates are near 2.0.

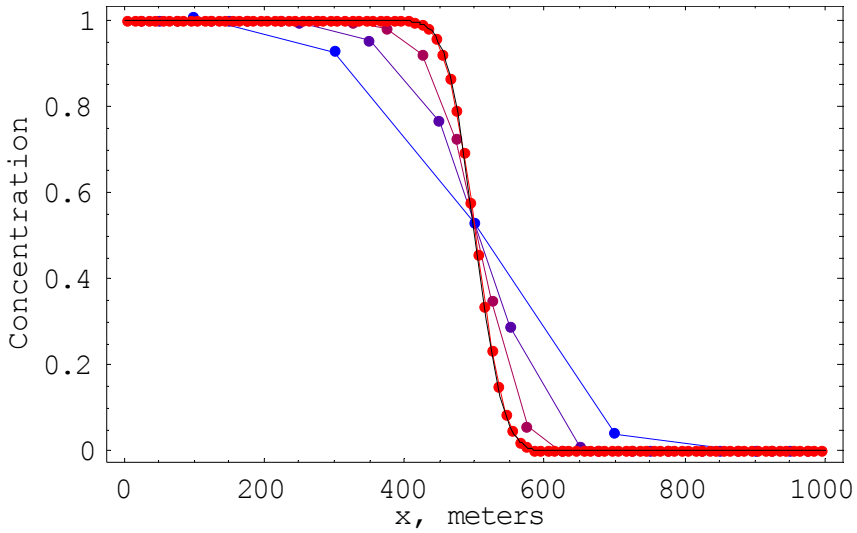


Figure 5.12 Simulation results for diffusive advection-dominated test case (N_x = 1.0 m^2/sec) at time t = 500 seconds. Grid resolution, dx [200 m to 10.0 m].

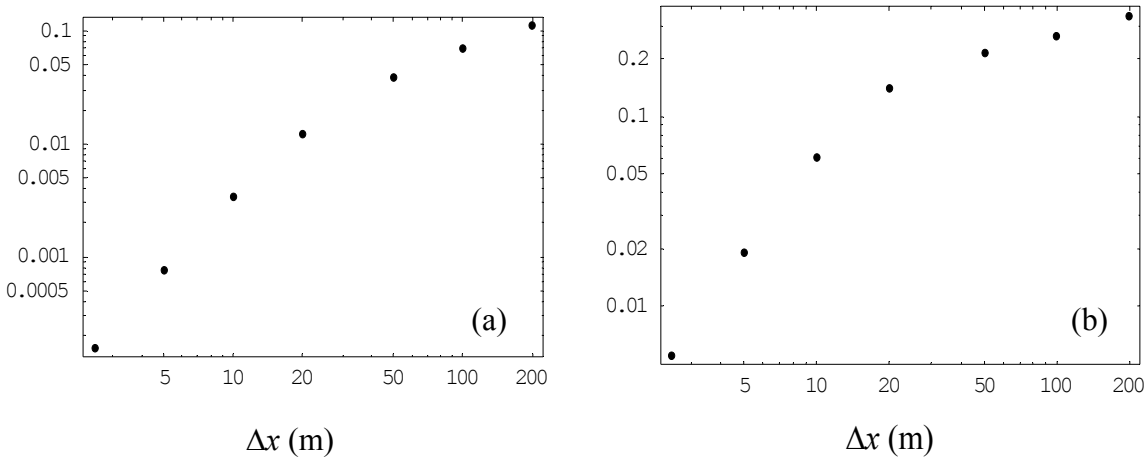


Figure 5.13 Log-log plot of spatial error norms for N_x = 1.0 m^2/sec : (a) L_2 errors and (b) L_{inf} errors.

Table 5.5 Grid convergence results for N_x = 1.0 m^2/sec.

Δx (m)	Pe_x	Cr_x	L_2	L_∞
200	200	0.0005	0.111453	0.341258
100	100	0.0001	0.071440	0.263432
50	50	0.0002	0.038416	0.216620
20	20	0.0005	0.012427	0.140579
10	10	0.001	0.003452	0.060299
5	5	0.002	0.000756	0.019279
2.5	2.5	0.004	0.000153	0.005450
Peak convergence rate:			2.302	1.823
Average convergence rate:			1.893	1.563
Best fit convergence rate:			1.863	1.571

N_x = 10.0 m^2/sec

Figures 5.14 and 5.15 show the simulation results and log-log plots of error for a more diffusive test case (Peclet numbers from 0.25 to 20). Table 5.6 presents a summary of the error norms. Notice that for the same resolution, this more diffusive test case has smaller

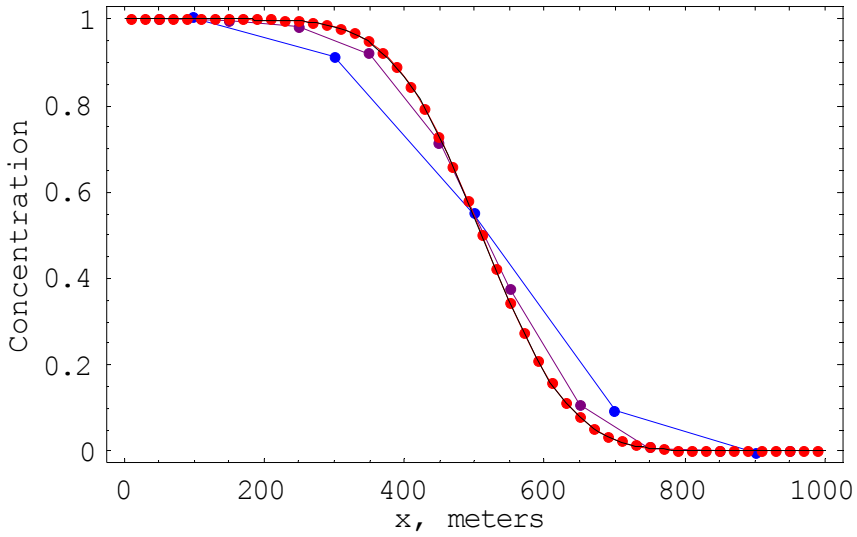


Figure 5.14 Simulation results for more diffusive test case (N_x = 10.0 m^2/sec) at time t = 500 seconds. Grid resolution, dx [200m to 20m].

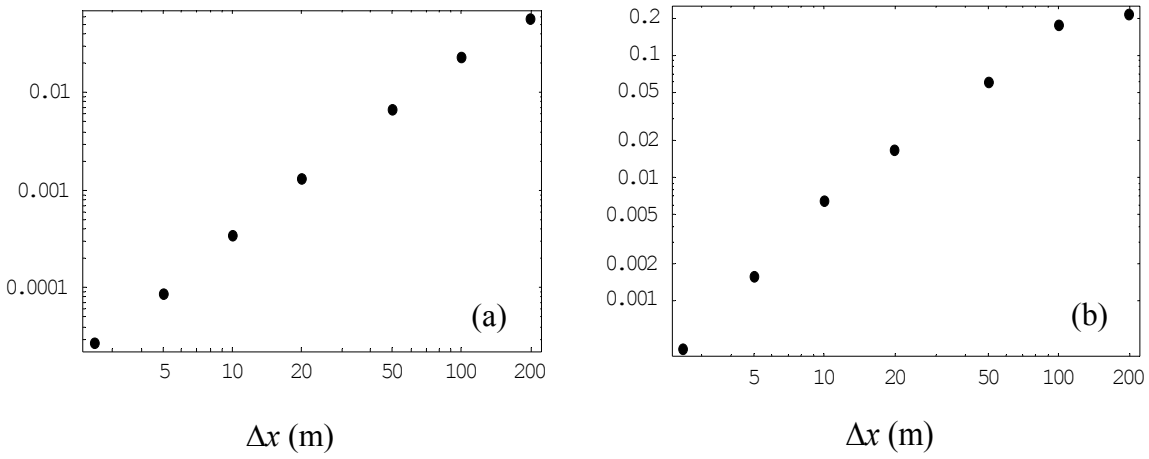


Figure 5.15 Log-log plot of spatial error norms for $N_x = 10.0$ m^2/sec : (a) L_2 errors and (b) L_{inf} errors.

Table 5.6 Convergence results for $N_x = 10.0$ m^2/sec.

Δx (m)	Pe_x	Cr_x	L_2	L_∞
200	20	0.0005	0.057080	0.214512
100	10	0.0001	0.023620	0.173952
50	5	0.0002	0.006881	0.060593
20	2	0.0005	0.001316	0.016845
10	1	0.001	0.000346	0.006499
5	0.5	0.002	0.000088	0.001567
2.5	0.25	0.004	0.000027	0.000407
Peak convergence rate:			1.982	2.052
Average convergence rate:			1.837	1.658
Best fit convergence rate:			1.853	1.615

errors than the previous test case, e.g., at $\Delta x = 20$m the L_2 error for $N_x = 1.0$ m^2/sec is 0.0124 while the L_2 error for $N_x = 10.0$ m^2/sec is 0.0013. Also, the errors are converging to zero and the convergence rates are approaching the theoretical value of 2.0.

$N_x = 25.0\ m^2/sec$

Figures 5.16 and 5.17 show the simulation results and log-log plots of error norms for a diffusion-dominated test case (Peclet numbers from 0.1 to 8). Table 5.7 presents a summary of the error norms. Notice that the smallest resolutions are not plotted in Figure

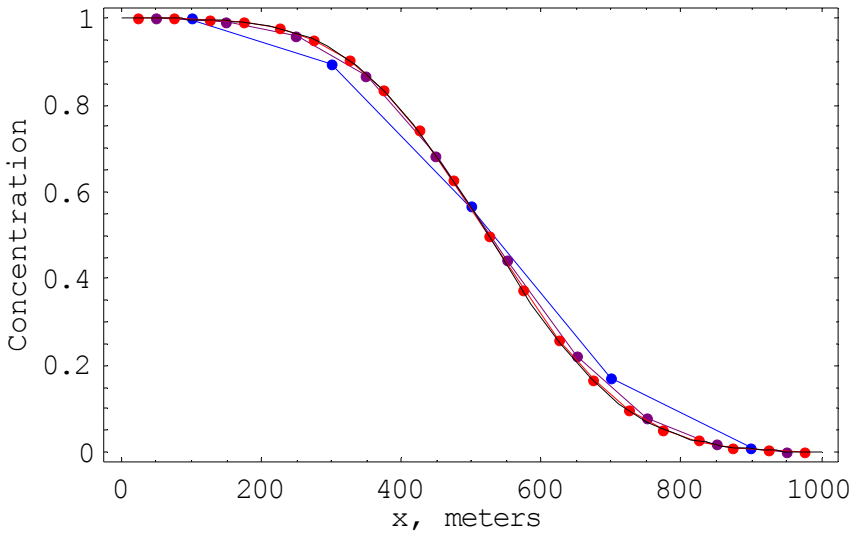


Figure 5.16 Simulation results for diffusion-dominated test case (N_x = 25.0 m^2/sec) at time t = 500 sec. Grid resolution, dx [200 m to 50 m].

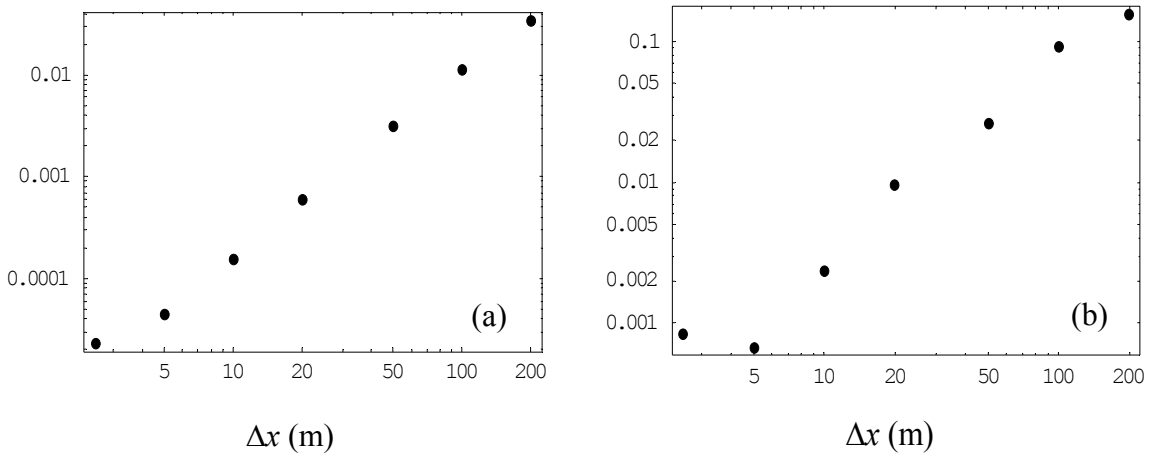


Figure 5.17 Log-log plot of spatial error norms for N_x = 25.0 m^2/sec : (a) L_2 errors and (b) L_{inf} errors.

Table 5.7 Grid convergence results for $N_x = 25.0$ m^2/sec.

Δx (m)	Pe_x	Cr_x	L_2	L_∞
200	8	0.00005	0.033998	0.154232
100	4	0.0001	0.011297	0.091014
50	2	0.0002	0.003122	0.026359
20	0.8	0.0005	0.000609	0.009585
10	0.4	0.001	0.000158	0.002340
5	0.2	0.002	0.000045	0.000675
2.5	0.1	0.004	0.000023	0.000846
Peak convergence rate:			1.950	2.035
Average convergence rate:			1.800	1.680
Best fit convergence rate:			1.814	1.602

5.16 since they lie on top of the red curve and make the plot too crowded. The peak convergence rates are near the theoretical value of 2.0 while the average and best-fit slopes are near the theoretical value.

Verification of *z*-coordinate direction

To set up a test case for the z coordinate, the horizontal velocity and diffusion coefficient, u and N_x, are set equal to zero and the vertical velocity, w, is set equal to 0.01. The solution at the middle of the channel is plotted after a simulated 1000 sec (when the plume has reached mid-depth) and then compared to the analytical solution given by,

$$C(z, t) = \frac{c_o}{2}\left[erfc\frac{(z - wt)}{2\sqrt{N_z t}} + \exp\frac{wz}{N_z} erfc\frac{(z + wt)}{2\sqrt{N_z t}}\right] \quad \textbf{(5.149)}$$

where *erfc* represents the complementary error function, c_o is the strength of the boundary condition, N_z is the diffusivity, z is the position along the channel depth, and t is the time. For this study, four diffusivities ranging from advection-dominated flows ($N_z = 10^{-6}$ m^2/ sec) to diffusion-dominated flows ($N_z = 0.004$ m^2/sec) are examined. The analytical solutions for the study diffusivities are shown in Figure 5.18.

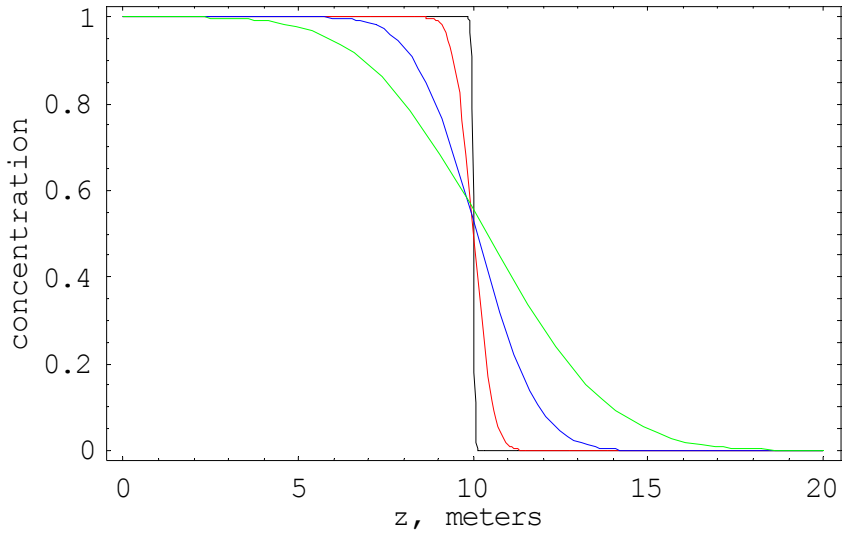


Figure 5.18 Analytical solution for advection-dispersion breakthrough curves in the z coordinate for $N_z = 10^{-6}$, 0.0001, 0.001 and 0.004 m^2/sec at $t = 1000$ seconds.

Again, the maximum cross-stream variation for the domain was monitored for each run and additional runs were made with increased resolution in the cross-stream (x) direction. As expected, the cross-stream variation was zero and no significant change was noticed in the error norms for this increased resolution, indicating that the solution is essentially one-dimensional in the z direction and the errors are dominated by the z grid resolution. For each diffusivity, plots of the simulated results for various grid spacings compared to the analytical solution and log-log plots of the error norms are presented, as well as a summary table of the convergence results.

$N_z = 10^{-6}\ m^2/sec$

Figures 5.19 and 5.20 show the simulation results and error norms for the advection-dominated test case (Peclet numbers from 625 to 40,000) and Table 5.8 presents a summary of the error norms. The simulation results for grid resolutions of $\Delta z = 0.125$m and 0.0625m are not shown since they lie between the two red curves and crowd the graph. Notice that the simulated results converge to the analytical solution as the grid is resolved.

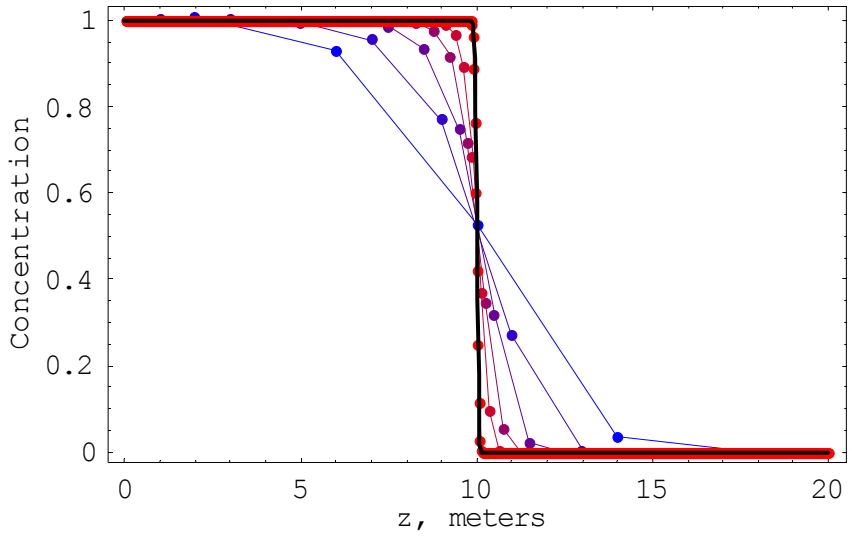


Figure 5.19 Simulation results for advection-dominated test case ($N_z = 10^{-6}$ m^2/sec) at time t = 1000 sec. Grid resolution, dz [4.0 m to 0.03125 m].

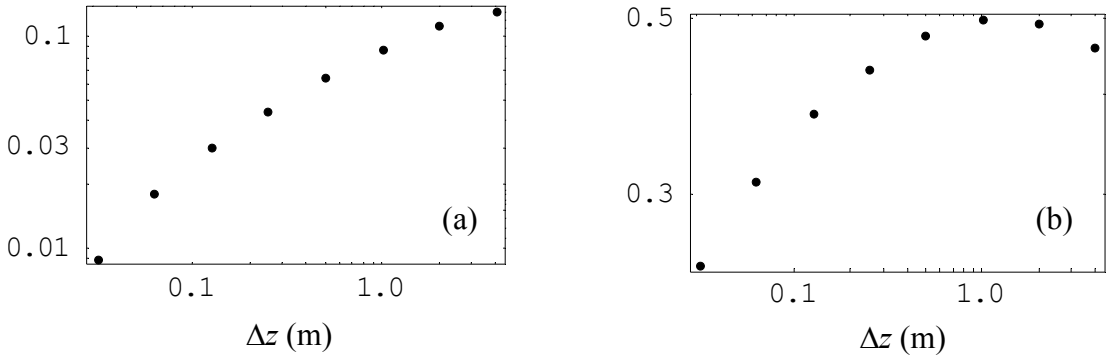


Figure 5.20 Log-log plot of spatial error norms for $N_z = 10^{-6}$ m^2/sec: (a) L_2 error, (b) L_{inf} error.

However, as was the case with the advection-dominated test case in the x coordinate, the L_∞ error does not approach zero because of the sharp front. As more data points are added to the portion of the curve where the front is located, the error will not decrease past a certain point due to the instantaneous jump of the analytical curve being approximated by discrete points. Also, the convergence rates are quite similar to those for the advection-

Table 5.8 Grid convergence results for $N_z = 10^{-6}$ m^2/sec.

Δz (m)	Pe_z	Cr_z	L_2	L_∞
4.0	40000	0.000025	0.13166755	0.45968364
2.0	20000	0.00005	0.11250578	0.49106329
1.0	10000	0.0001	0.08694574	0.49701228
0.5	5000	0.0002	0.06409106	0.47541374
0.25	2500	0.0004	0.04489673	0.42970364
0.125	1250	0.0008	0.02997586	0.37999885
0.0625	625.	0.0016	0.01821482	0.31020307
0.03125	312.5	0.0032	0.00889667	0.24381869
Peak convergence rate:			1.034 (0.556)*	0.347 (0.385)
Average convergence rate:			0.555 (0.394)	0.205 (0.115)
Best fit convergence rate:			0.541 (0.402)	0.205 (0.075)

* Numbers in parentheses are the *x* convergence rates from Table 5.4.

dominated test case in the *x*-coordinate direction, given in parentheses in Table 5.8. Finally, due to the nature of the sharp front, the convergence rates are well below the theoretical value of 2.0 for linear interpolants on smooth solutions.

N_z= 0.0001 m^2/sec

Figures 5.21 and 5.22 show the simulation results and log-log plots of error norms for a more diffusive advection-dominated test case (Peclet numbers from 6.25 to 400) and Table 5.9 presents a summary of the error norms. As the grid is resolved, the model results converge to the analytical solution. The simulation results for the finest resolution are not shown since they overlie the red curve and crowd the graph. The peak convergence rate for the L_2 error norm is near the theoretical value of 2.0 for linear interpolants, but the rates for the L_∞ error norm are somewhat lower. The convergence rates in the *z* coordinate are also similar to the coordinating test case in the *x* coordinate (N_x = 1.0 m^2/sec), shown in parentheses in Table 5.9.

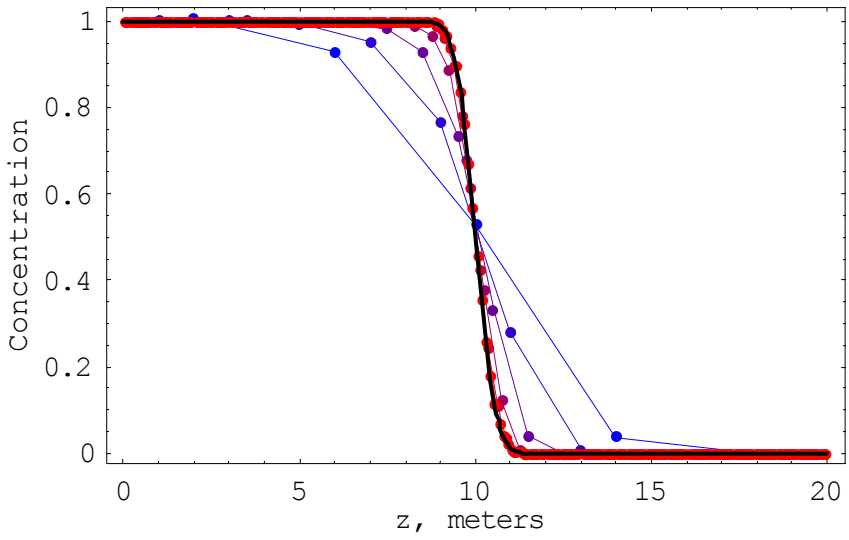


Figure 5.21 Simulation results for slightly diffusive advection-dominated test case ($N_z = 0.0001$ m^2/sec) at time $t =$ 1000 sec. Grid resolution, Δz [4.0 m to 0.125 m].

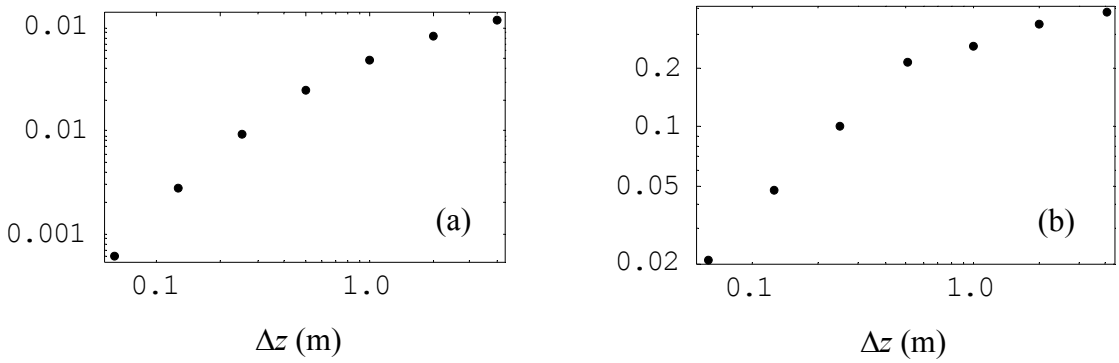


Figure 5.22 Log-log plot of spatial error norms for $N_z = 0.0001$ m^2/sec: (a) L_2 error, (b) L_{inf} error.

Table 5.9 Grid convergence results for N_z = 0.0001 m^2/sec.

Δz (m)	Pe_z	Cr_z	L_2	L_∞
4.0	400	0.000025	0.12098928	0.38867371
2.0	200	0.00005	0.08355729	0.34125833
1.0	100	0.0001	0.05026854	0.26343150
0.5	50	0.0002	0.02469958	0.21662041
0.25	25	0.0004	0.00940659	0.10147721
0.125	12.5	0.0008	0.00278536	0.04774518
0.0625	6.25	0.0016	0.00060743	0.02109214
Peak convergence rate:			2.197 (2.302)*	1.179 (1.823)
Average convergence rate:			1.782 (1.893)	1.120 (1.563)
Best fit convergence rate:			1.779 (1.863)	1.117 (1.571)
* Numbers in parentheses are the x convergence rates from Table 5.5.				

N_z= 0.001 m^2/sec

Figures 5.24 and 5.23 show the log-log plots of error norms and the simulation results for a more diffusive test case (Peclet numbers from 0.3125 to 40). Table 5.10 presents a summary of the error norms. Breakthrough curves for resolutions smaller than

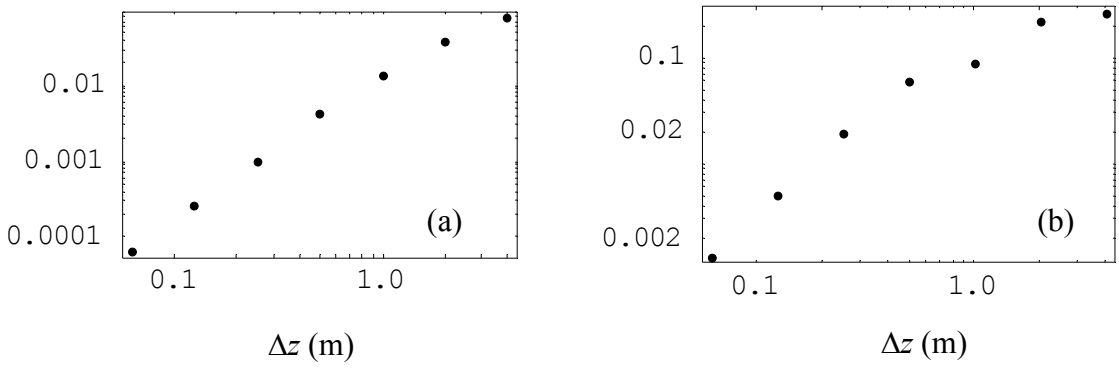


Figure 5.23 Log-log plot of spatial error norms for N_z = 0.001 m^2/sec: (a) L_2 error, (b) L_{inf} error.

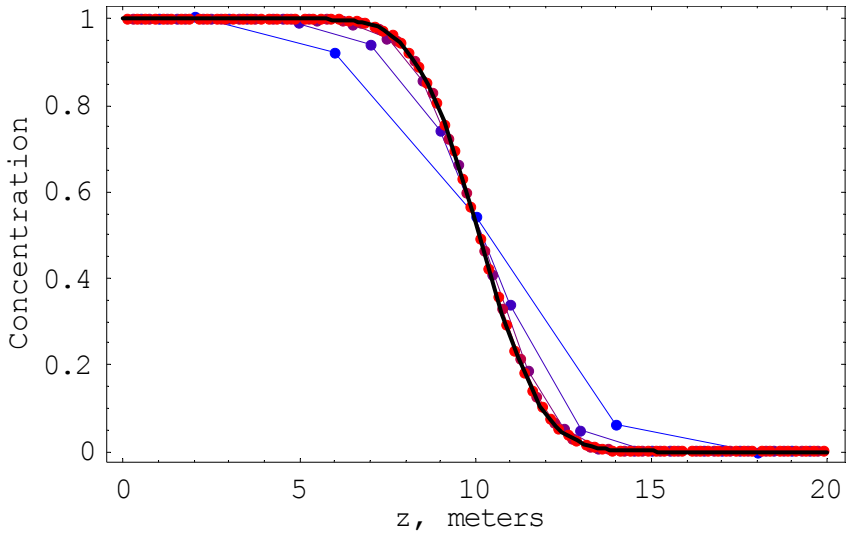


Figure 5.24 Simulation results for diffusive test case (N_z = 0.001 m^2/sec) at time t = 1000 sec. Grid resolution, dz [4.0 m to 0.25 m].

Table 5.10 Grid convergence results for N_z = 0.001 m^2/sec.

Δz (m)	Pe_z	Cr_z	L_2	L_∞
4.0	40	0.000025	0.07626821	0.26002591
2.0	20	0.00005	0.03710528	0.21451225
1.0	10	0.0001	0.01295931	0.08672942
0.5	5	0.0002	0.00410147	0.06059333
0.25	2.5	0.0004	0.00098905	0.01936984
0.125	1.25	0.0008	0.00025323	0.00511611
0.0625	0.625	0.0016	0.00006376	0.00132384
Peak convergence rate:			2.052 (1.982)*	1.950 (2.052)
Average convergence rate:			1.917 (1.837)	1.839 (1.658)
Best fit convergence rate:			1.935 (1.853)	1.847 (1.615)
* Numbers in parentheses are the *x* convergence rates from Table 5.6.				

Δz = 0.25m are not shown in Figure 5.24 since they lie on top of the red curve. Notice that the model results converge to the analytical solution. The peak and best-fit convergence rates are near the theoretical value of 2.0 for both error norms; and the rates are similar to

the corresponding x coordinate test case with N_x = 10.0 m^2/sec, given in parentheses.

N_z= 0.004 m^2/sec

Figures 5.25 through 5.26 show the simulation results and log-log plots of error norms for the most diffusive test case (Peclet numbers from 0.15625 to 10) and Table 5.11 presents a summary of the error norms. Breakthrough curves for resolution smaller than Δz

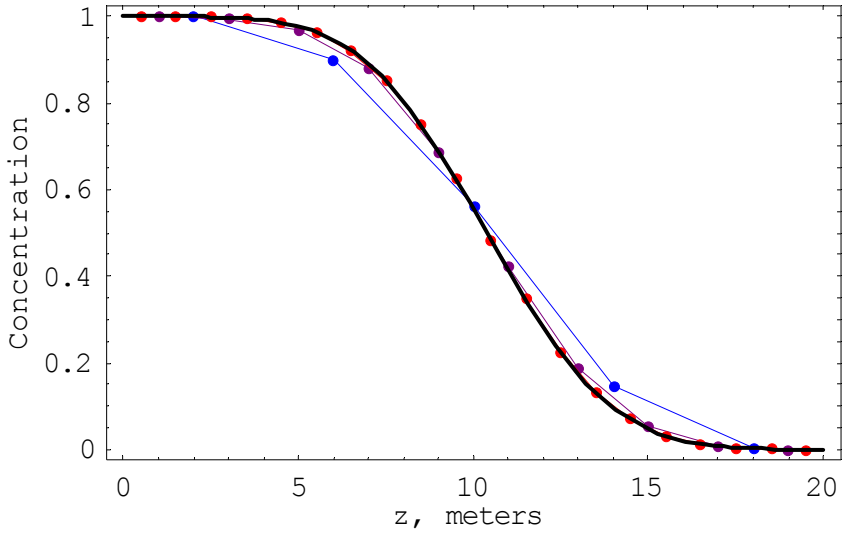


Figure 5.25 Simulation results for diffusion-dominated test case (N_z = 0.004 m^2/sec) at time t = 1000 sec. Grid resolution, dz [4.0 m to 1.0 m].

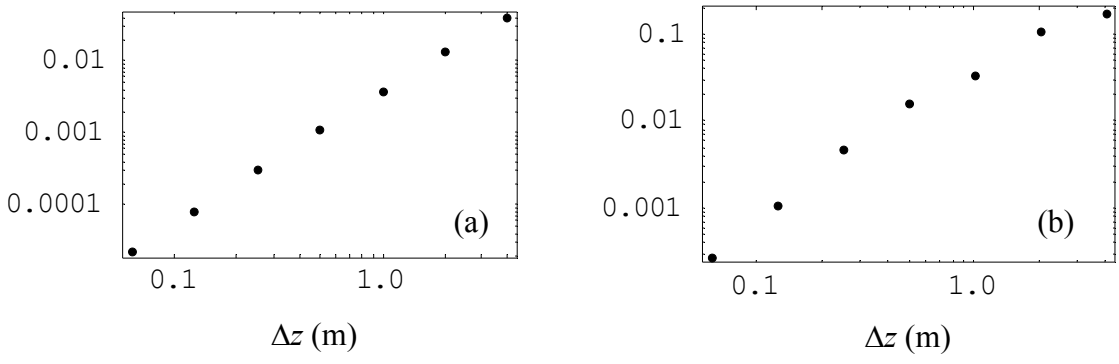


Figure 5.26 Log-log plot of error norms for N_z = 0.004 m^2/sec: (a) L_2 error and (b) L_{inf} error.

Table 5.11 Grid convergence results for $N_z = 0.004$ m^2/sec.

Δz (m)	Pe_z	Cr_z	L_2	L_∞
4.0	10	0.000025	0.03897427	0.17394397
2.0	5	0.00005	0.01365194	0.11037563
1.0	2.5	0.0001	0.00372179	0.03296036
0.5	1.25	0.0002	0.00113126	0.01642067
0.25	0.625	0.0004	0.00029712	0.00475637
0.125	0.3125	0.0008	0.00008002	0.00107441
0.0625	0.15625	0.0016	0.00002246	0.00027431
Peak convergence rate:			1.929 (1.950)*	2.146 (2.035)
Average convergence rate:			1.794 (1.800)	1.551 (1.680)
Best fit convergence rate:			1.813 (1.814)	1.574 (1.602)
* Numbers in parentheses are the x convergence rates from Table 5.7.				

= 1.0m are not shown in Figure 5.25 since they lie nearly over the red curve and make the graph too crowded. For this diffusion dominated test case, the transport model converges to the analytical solution. The peak convergence rates for both the error norms are near the theoretical value of 2.0; and all of the rates are similar to the convergence rates for the diffusion dominated test case in the x-direction ($N_x = 25.0$ m^2/sec), given in parentheses.

The LDG method presented herein performs quite well over the range of test cases in both spatial directions: from advection dominated to diffusion dominated. Temporal convergence results for the validation test cases will be presented in §5.7.3 and mass balance tests for a simple Gauss plume are presented in the next section.

5.7.2 Examination of mass balance

To evaluate this algorithm's ability to preserve mass, a test case was run with a known initial mass. The total mass in the domain was computed and compared to this initial mass as the simulation progressed. The initial condition was a Gaussian distribution in either the x or z coordinate. Constant grid spacing and velocity fields were used, and the cross-stream diffusion and velocity were set to zero to simulate 1D transport in the direction

of interest. The simulations were run long enough to transport the distribution a significant distance without any of the plume leaving the domain. Also, the dimension for the coordinate of interest was doubled, relative to the verification test cases of §5.7.1, to ensure that the plume remains inside the domain of interest. Type II boundary conditions with an inflow concentration of 0.0 were used for both the *x* and *z* coordinate directions. Total mass was calculated by multiplying each elemental average concentration by the corresponding elemental area and summing these quantities over the entire domain.

Grid convergence results were generated for all of the diffusivity values used in the 1D validation studies. As in the validation studies, the L_2 and L_∞ error norms were calculated for the total mass over space and time. Graphs of the total mass to initial mass ratio, which should equal 1.0 for all time if the algorithm is conservative, are presented for select grid resolutions. Log-log plots are given only for the L_2 error norm, as the L_∞ error norms have similar behavior and differ only in magnitude. Both error norms are summarized in tabular form.

Analysis of the *x*-coordinate direction

The domain of interest was a 2000m channel with constant 20m bathymetry. The vertical grid spacing was held constant at $\Delta z = 2.0$m. The Gaussian distribution was initialized in the *x* coordinate and was constant with depth. The initial plume was centered at x=100.0m and the simulation was run for 1000sec with a velocity of u=1.0 m/s. Thus, the final distribution should be centered at x=1100.0m. The Gaussian spread was σ=19.9471, which gives an integrated area of 50.0 moles/m^2 over the length of the channel. When multiplied by the bathymetry (and a unit width), this yields a total initial mass of exactly 1000.0 moles. This test case was run for the four values of diffusivity presented above in the 1D validation, N_x = 0.001, 1.0, 10.0 and 25.0 m^2/s. To ensure that any errors in mass are dominated by the spatial resolution, a time step of $\Delta t = 0.001$ was used for all of the mass balance runs in the *x* coordinate. Temporal resolution for accuracy and mass balance will

be discussed further in §5.7.3.

The effect of boundary conditions on mass balance was examined for two of the diffusivities, N_x = 1.0 and 25.0 m^2/s. For each of these test cases, all three boundary types were examined for the x coordinate while the BCs for the z coordinate remained fixed at Type II (since w=0.0, this results in Type III BCs in z). For Type I BCs the condition was set as $\tilde{c} = 0.0$ so that no new mass was added to the system. For all boundary conditions, the time step was 0.001 sec.

N_x= 0.001 m^2/s

Figure 5.27 shows the mass to initial mass ratios, the L_2 error norm, and the Gauss plumes for the advection-dominated test case. Table 5.12 presents a summary of the error norms. From the tabulated errors, note that the odd behavior for the coarsest grid is in the sixth decimal place. For all but the coarsest grid resolution, the mass balance for this

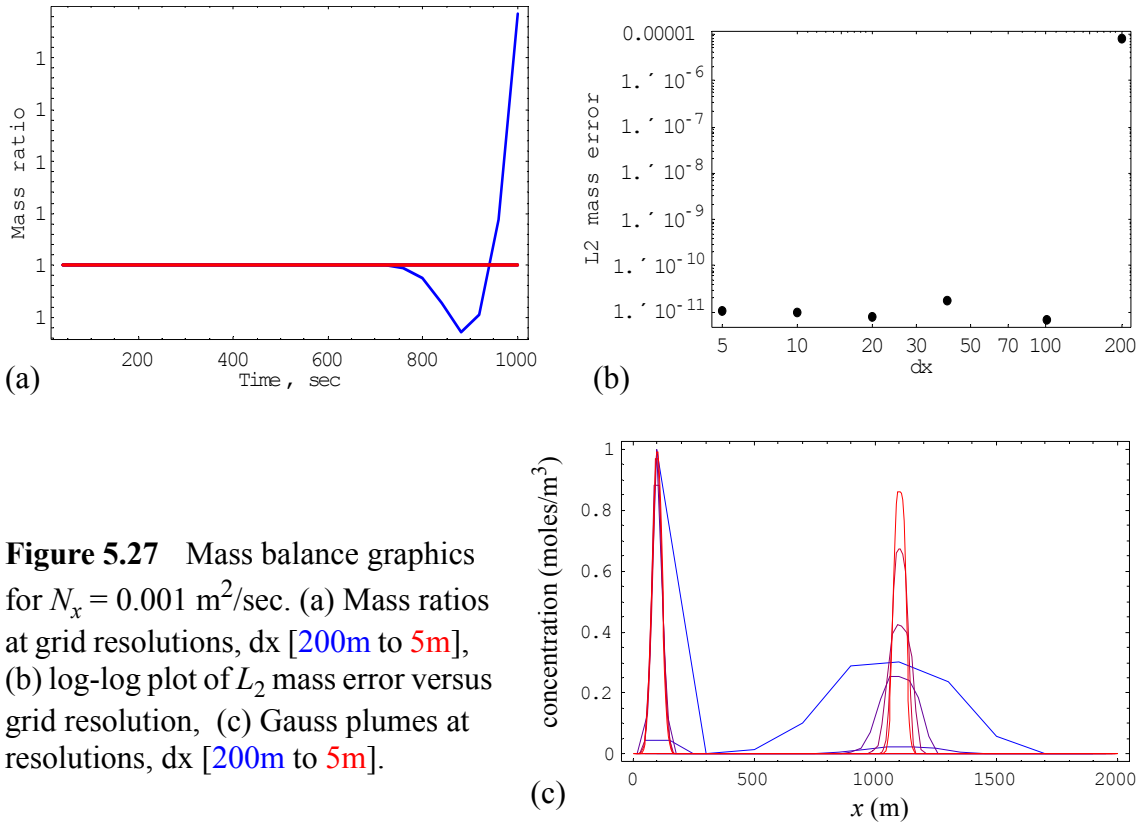


Figure 5.27 Mass balance graphics for N_x = 0.001 m^2/sec. (a) Mass ratios at grid resolutions, dx [200m to 5m], (b) log-log plot of L_2 mass error versus grid resolution, (c) Gauss plumes at resolutions, dx [200m to 5m].

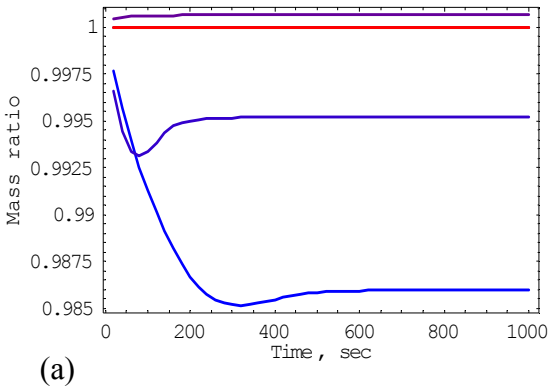
Table 5.12 Spatial errors in mass balance for N_x = 0.001 m^2/s.

Δx (m)	Pe_x	Cr_x	L_2	L_∞
200	200000	0.000005	8.3768E-06	3.8770E-05
100	100000	0.00001	6.7287E-12	9.8339E-12
40	40000	0.000025	1.7995E-11	3.4220E-11
20	20000	0.00005	8.3099E-12	2.1259E-11
10	10000	0.0001	9.7435E-12	1.9895E-11
5	5000	0.0002	1.1241E-11	2.1487E-11

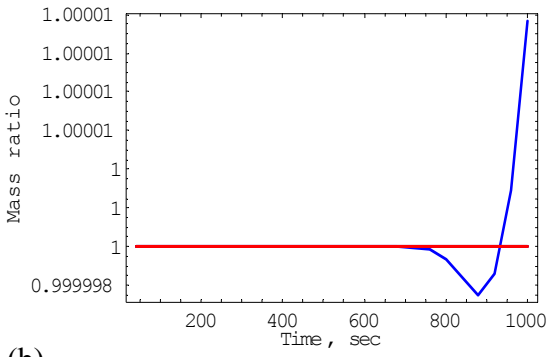
advective test case is quite good. Note that for the most coarse resolutions, the initial Gauss plume is not well represented with so few data points and the peak may not be captured, such that the initial mass is not equal to exactly 1000 moles. Despite this difference in initial mass, however, the algorithm is able to conserve the total mass in the domain with any spatial resolution.

N_x= 1.0 m^2/sec

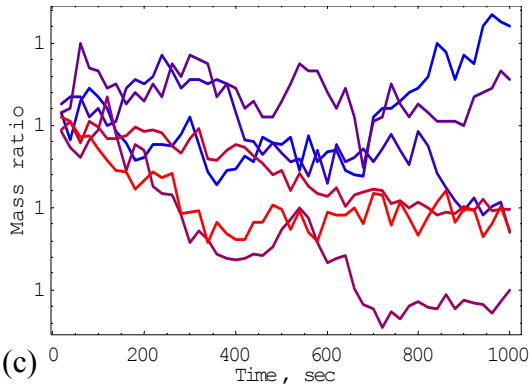
Figures 5.28 through 5.30 compare the mass ratios, the initial and final Gauss plumes, and the L_2 mass errors for a more diffusive advection-dominated test case with all three types of boundary conditions. Table 5.13 presents a summary of the error norms. Note that the oscillatory nature of the mass ratio for Type III BCs is on a scale of 10^{-12}. For both the natural, or Type II, and no-flux Type III boundary conditions, the total mass in the system is conserved as the grid is resolved. Again, the coarsest resolutions do not accurately capture the initial Gaussian distribution, but mass balance is not too bad at any of the spatial resolutions. For this slightly diffusive test case, perfect mass balance can be achieved with almost any resolution with Type II or III BCs. However, for Type I BCs only the fine spatial resolutions have acceptable mass errors, and even then perfect mass balance cannot be achieved. This will be discussed further when the results for the diffusion dominated test case are presented below. Notice also that for the three boundary types the final Gauss plumes themselves are only visibly different at the coarsest resolutions.



(a)



(b)



(c)

Figure 5.28 Mass ratio versus time for N_x = 1.0 m^2/sec with: (a) Type I, (b) Type II, and (c) Type III BCs. Grid resolutions, dx [200m to 5m].

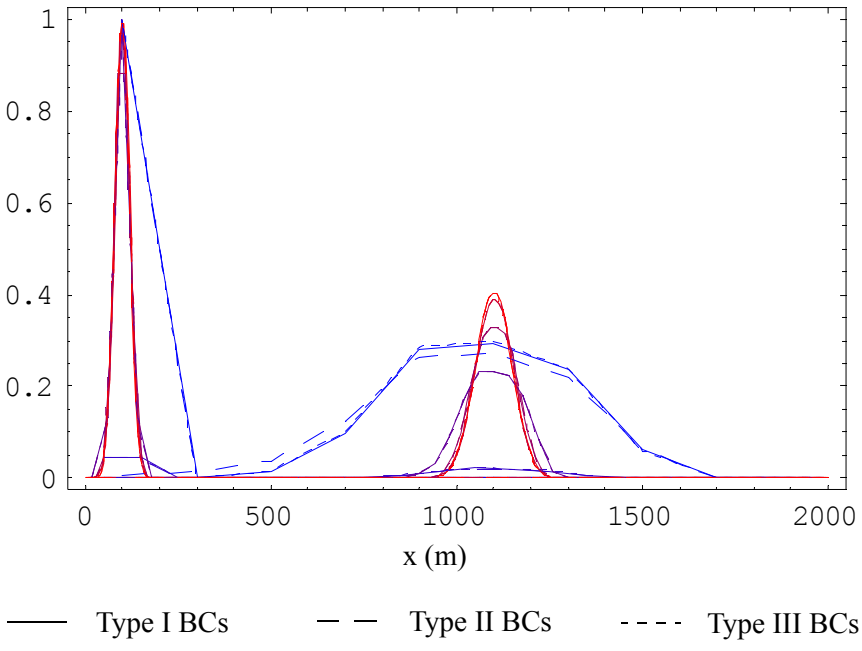


Figure 5.29 Gauss plumes for N_x = 1.0 m^2/sec with grid resolutions, dx [200m to 5m].

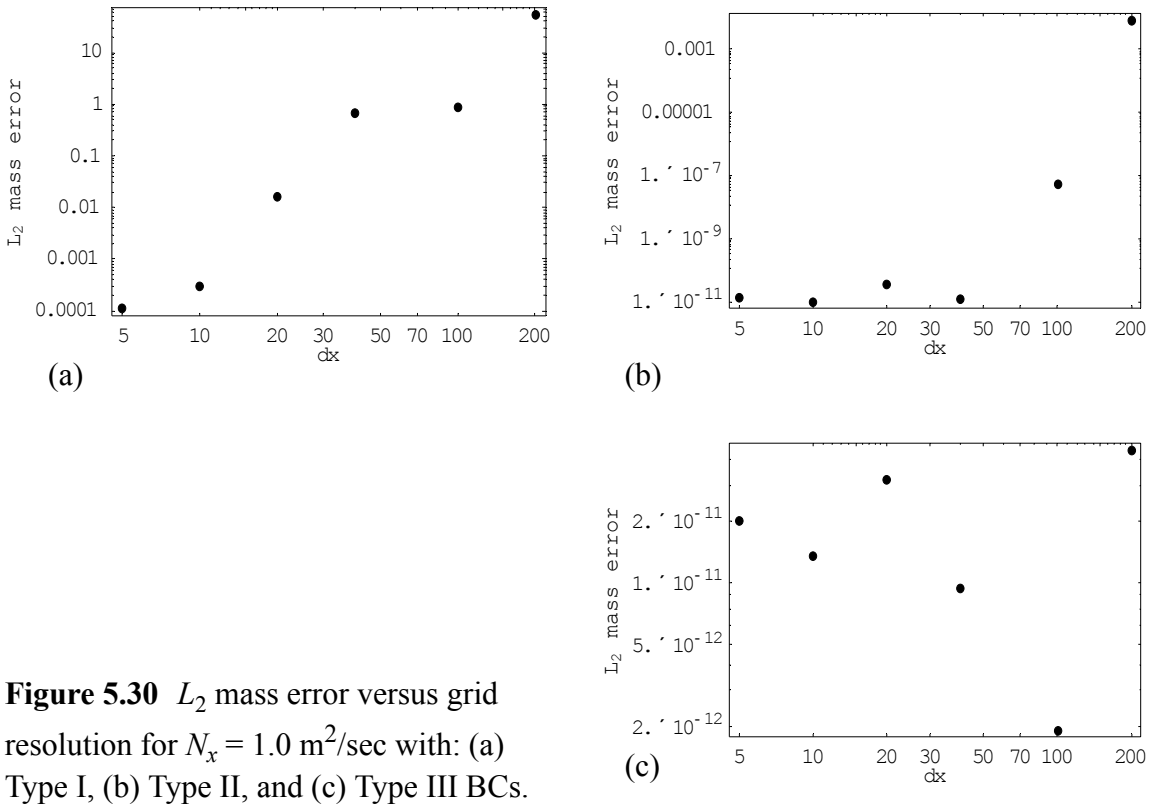


Figure 5.30 L_2 mass error versus grid resolution for N_x = 1.0 m^2/sec with: (a) Type I, (b) Type II, and (c) Type III BCs.

Table 5.13 Spatial errors in mass balance for N_x = 1.0 m^2/s.

Δx (m)	Pe_x	Cr_x	Type I BCs		Type II BCs		Type III BCs	
			L_2	L_∞	L_2	L_∞	L_2	L_∞
200	200	0.000005	53.56460	59.44260	0.008205	0.046605	4.4261e-11	1.0914e-10
100	100	0.00001	0.860520	1.188620	5.2653e-8	2.7574e-7	1.9160e-12	4.4906e-12
40	40	0.000025	0.654976	0.663270	1.2311e-11	2.1146e-8	9.4109e-12	2.0464e-11
20	20	0.00005	0.016647	0.016689	3.7624e-11	5.5024e-11	3.2056e-11	4.9113e-11
10	10	0.0001	0.000309	0.000310	1.0059e-11	1.6939e-11	1.3607e-11	2.1828e-11
5	5	0.0002	0.000108	0.000110	1.3522e-11	2.2055e-11	2.0246e-11	2.8422e-11

N_x= 10.0 m^2/sec

Figure 5.31 shows the mass ratios, the L_2 mass errors and the Gauss plumes for a diffusion-dominated test case. Table 5.14 presents a summary of the error norms. As the

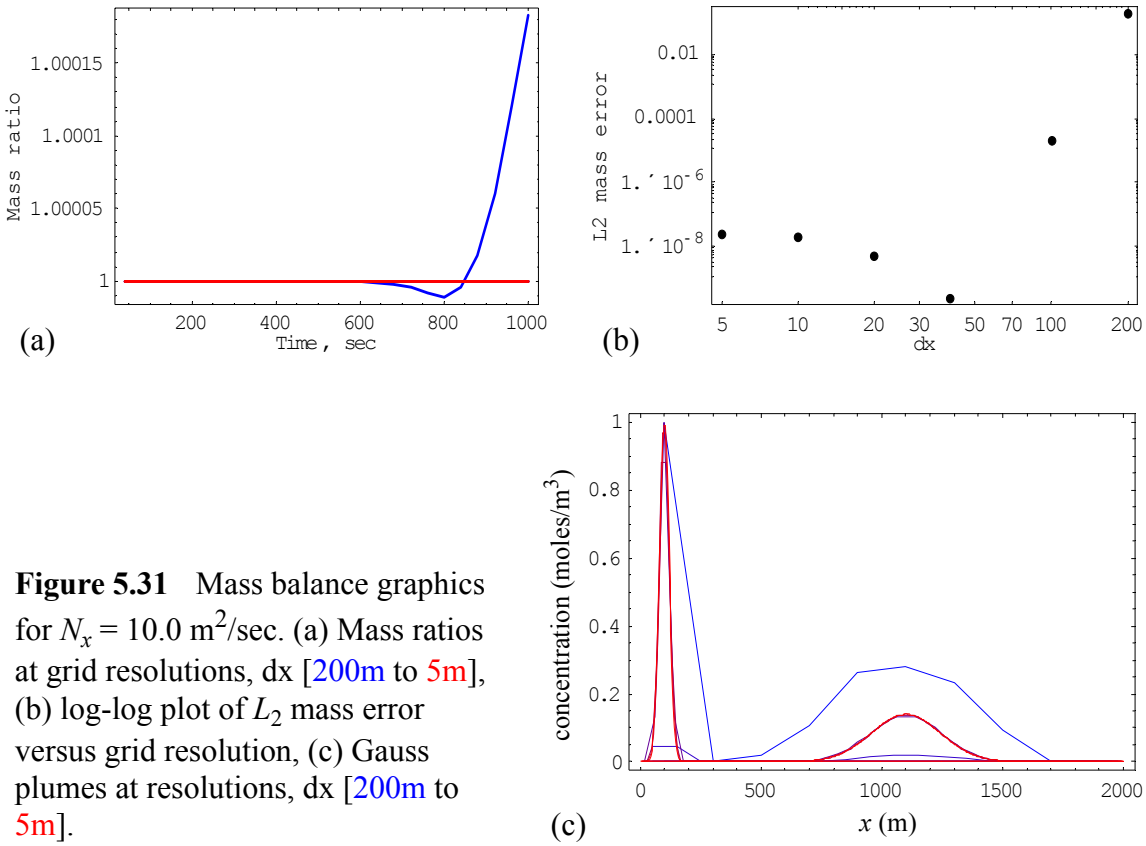


Figure 5.31 Mass balance graphics for N_x = 10.0 m^2/sec. (a) Mass ratios at grid resolutions, dx [200m to 5m], (b) log-log plot of L_2 mass error versus grid resolution, (c) Gauss plumes at resolutions, dx [200m to 5m].

Table 5.14 Spatial errors in mass balance for N_x = 10.0 m^2/s.

Δx (m)	Pe_x	Cr_x	L_2	L_∞
200	20	0.000005	0.182358	0.730626
100	10	0.00001	2.1258E-5	9.0626E-5
40	4	0.000025	2.4902E-10	1.2193E-9
20	2	0.00005	5.0137E-9	2.5058E-8
10	1	0.0001	1.9285E-8	9.6275E-8
5	0.5	0.0002	2.4657E-8	1.2305E-7

grid is resolved, the total mass in the system is conserved and the mass ratio approaches unity. This is evident in the mass ratio plots and the error norms. With enough resolution, near perfect mass balance can be achieved for this diffusion-dominated case.

$N_x = 25.0\ m^2/sec$

Figures 5.32 through 5.34 compare the initial and final Gauss plumes, the mass to initial mass ratios, and L_2 mass errors for the most diffusive test case with all three types of boundary conditions. Table 5.15 presents a summary of the error norms. Note that the scale for the oscillations in mass ratio for Type III BCs is on the order of 10^{-12}.

For this highly diffusive test case, mass balance is not achieved with Type I boundary conditions. In fact, as the grid is resolved, the average error in the domain converges to a value of about 25 moles, meaning that the total mass in the system is approaching 975 instead of 1000. This error is significant enough that it is evident in the mass ratio as well. However, by using Type II boundary conditions much better mass balance can be achieved. The average error converges to a value around 0.005, which is significantly better than the previous error of 24 but is not as good as can be achieved with lower values of the diffusion coefficient. This error in mass can be explained by noticing that there is a decline in the mass ratio as the final time is approached. Due to the large value of diffusion, the outer edge of the plume has reached the right boundary by t=1000 sec. With Type II BCs, the plume is allowed to leave the domain and therefore mass is leaving the system. With Type III BCs, the boundary is closed so that mass cannot escape and

Table 5.15 Spatial errors in mass balance for N_x = 25.0 m^2/s.

			Type I BCs		*Type II BCs*		*Type III BCs*	
Δx (m)	Pe_x	Cr_x	L_2	L_∞	L_2	L_∞	L_2	L_∞
200	8	0.00005	927.546	982.25	0.406721	1.43226	1.6670e-10	2.2146e-10
100	4	0.0001	22.9499	23.3934	6.9206E-5	3.3421E-4	1.6612e-12	3.1264e-12
40	1.6	0.00025	22.0599	23.4249	0.004178	0.020108	8.3451e-12	1.6371e-11
20	0.8	0.0005	22.8584	24.2385	0.005359	0.025555	5.0720e-12	1.1141e-11
10	0.4	0.001	23.5706	24.9878	0.005652	0.026895	5.4143e-12	1.3074e-11
5	0.2	0.002	23.7049	25.1323	0.005727	0.027239	6.1926e-12	1.3188e-11

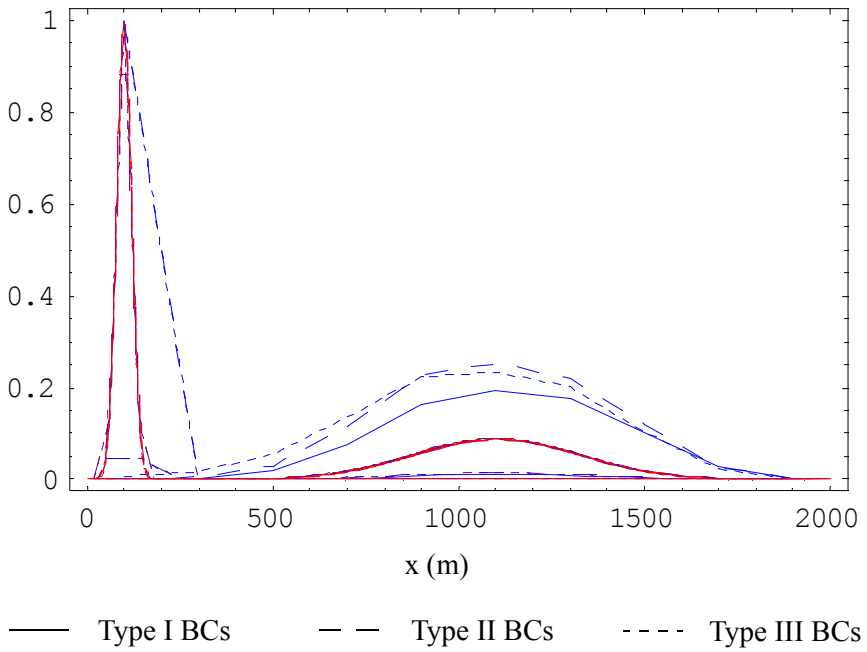


Figure 5.32 Gauss plumes for N_x = 25.0 m^2/sec with grid resolutions, dx [200m to 5m].

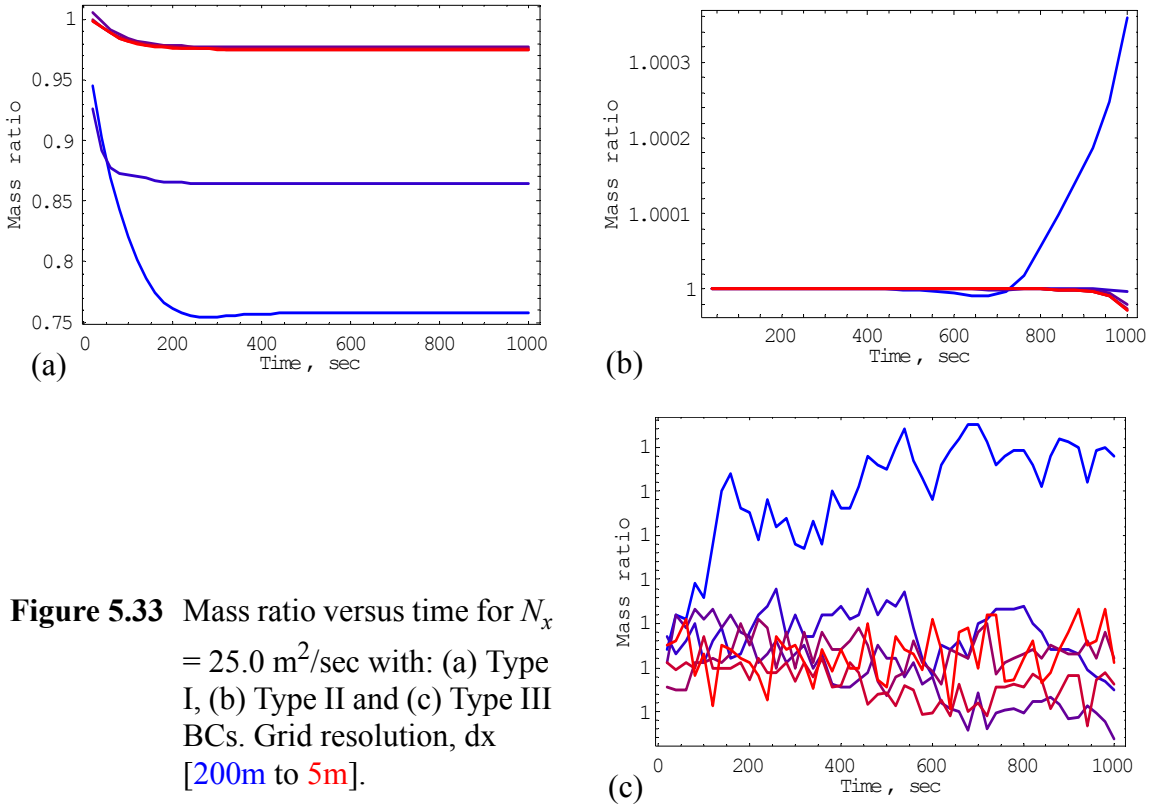


Figure 5.33 Mass ratio versus time for N_x = 25.0 m^2/sec with: (a) Type I, (b) Type II and (c) Type III BCs. Grid resolution, dx [200m to 5m].

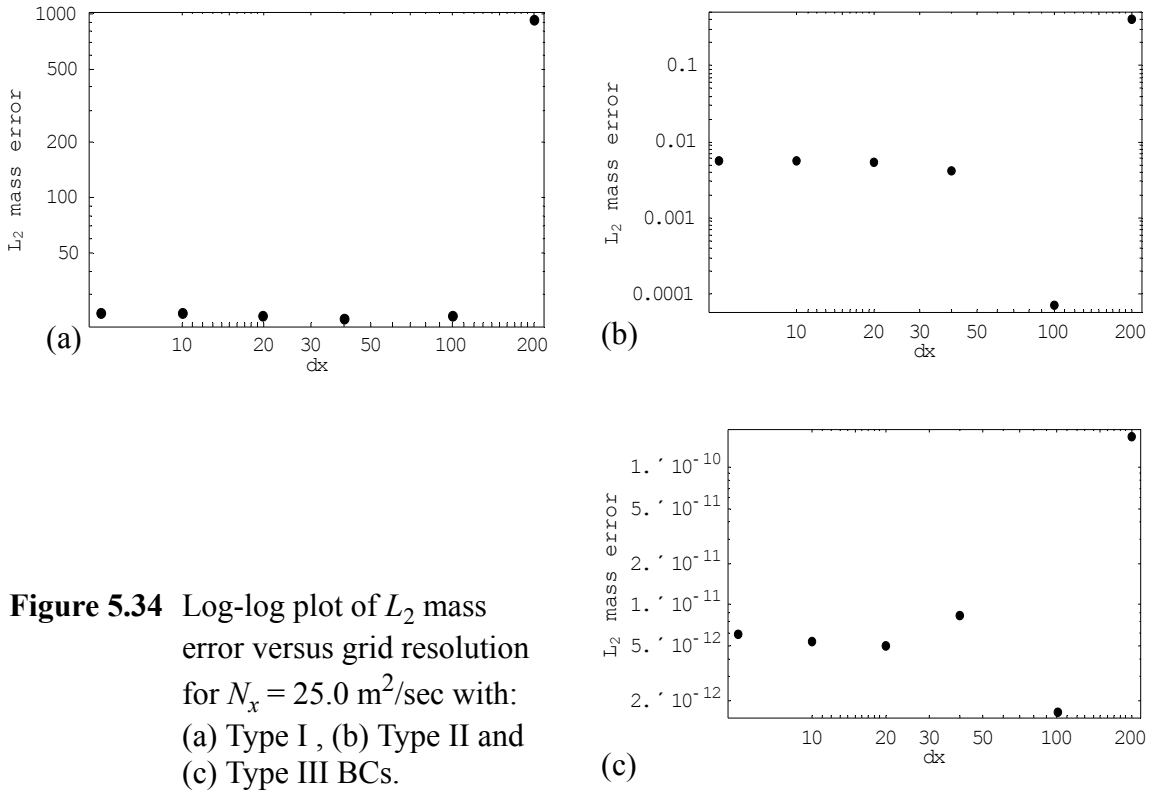


Figure 5.34 Log-log plot of L_2 mass error versus grid resolution for N_x = 25.0 m^2/sec with: (a) Type I , (b) Type II and (c) Type III BCs.

perfect mass balance is achievable at any resolution. Notice also that for the three boundary types the final Gauss plumes themselves are only visibly different at the coarsest resolutions. Mass balance and boundary conditions are discussed further in §5.7.3 when the temporal resolution is examined more closely.

Analysis of the *z*-coordinate direction

The domain of interest was a 1000m channel with constant 40m bathymetry. The horizontal grid spacing was held constant at Δx = 100.0m. The Gaussian distribution was initialized in the *z* coordinate and was constant along the channel. The initial plume was centered at *z*=10.0m and the simulation was run for 1000sec with a velocity of *w*=0.01 m/s. Thus, the final distribution should be centered at *z*=20.0m. A time step of Δt = 0.01sec was sufficient to maintain mass balance, and no significant change was noticed when a smaller time step was used. The Gaussian spread was σ=1.196845, which gives an

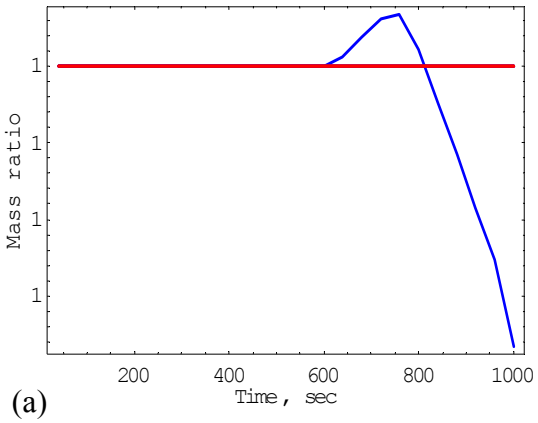
integrated area of 3.0 moles/m^2 over the depth of the channel. When multiplied by the channel length (and a unit width), this yields a total initial mass of 3000.0 moles. This test case was run for the four values of diffusivity presented above in the 1D validation, N_z = 0.000001, 0.0001, 0.001 and 0.005 m^2/s. Only Type II boundary conditions were tested in the *z* coordinate, since the results with Type I or III would be similar to what was found in the *x* coordinate.

$N_z = 10^{-6}\ m^2/sec$

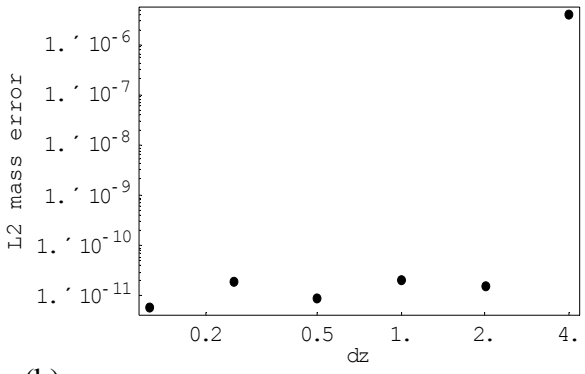
Figure 5.35 shows the mass to initial mass ratios, the L_2 mass errors, and the initial and final Gauss plumes for the advection-dominated test case and Table 5.16 presents a summary of the error norms. Note that the differences in the scale for Figure 5.35a are in the eleventh decimal place, such that all of the mass ratios are essentially equal to one. For this advection-dominated test case, perfect mass balance can be achieved with nearly all of the resolutions. As in the *x* coordinate test cases, the coarsest resolutions do not capture the peak of the Gaussian distribution, so the initial mass is not equal to exactly 3000 moles; however, the initial mass is still conserved. This is evident in the plot of mass ratios and in the convergence results.

Table 5.16 Spatial errors in mass balance for N_z = 0.000001 m^2/s.

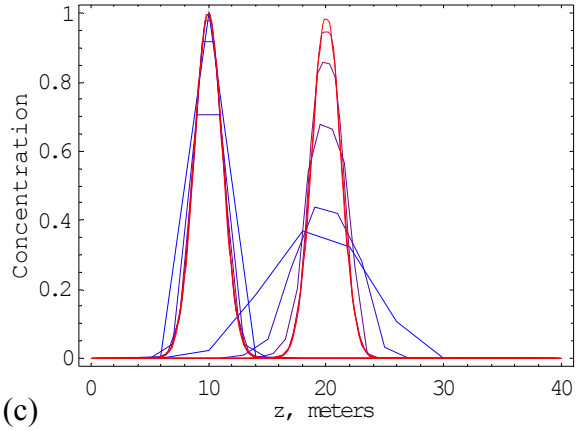
Δz (m)	Pe_z	Cr_z	L_2	L_∞
4	40000	0.000025	4.0898E-6	0.00001473
2	20000	0.00005	1.5464E-11	3.8654E-11
1	10000	0.0001	2.1243E-11	3.3651E-11
0.5	5000	0.0002	9.0054E-12	1.8645E-11
0.25	2500	0.0004	1.9038E-11	2.8649E-11
0.125	1250	0.0008	5.9327E-12	1.2733E-11



(a)



(b)



(c)

Figure 5.35 Mass balance graphics for $N_z = 10^{-6}$ m^2/sec. (a) Mass ratios at grid resolutions, dz [4m to 0.125m], (b) log-log plot of L_2 mass error versus grid resolution, (c) Gauss plumes at resolutions, dz [4m to 0.125m].

N_z= 0.0001 m^2/sec

Figure 5.36 shows the mass ratios, the L_2 mass errors, and the initial and final Gauss plumes for a more diffusive advection-dominated test case, and Table 5.17 presents a summary of the error norms. For this more diffusive advection-dominated test case, perfect

Table 5.17 Spatial errors in mass balance for N_z = 0.0001 m^2/s.

Δz (m)	Pe$_z$	Cr$_z$	L_2	L_∞
4	400	0.000025	0.00040817	0.00190551
2	200	0.00005	1.2259E-8	4.7407E-8
1	100	0.0001	2.0488E-11	3.7289E-11
0.5	50	0.0002	1.5217E-11	2.6830E-11
0.25	25	0.0004	8.9607E-12	1.9554E-11
0.125	12.5	0.0008	5.5397E-12	1.2278E-11

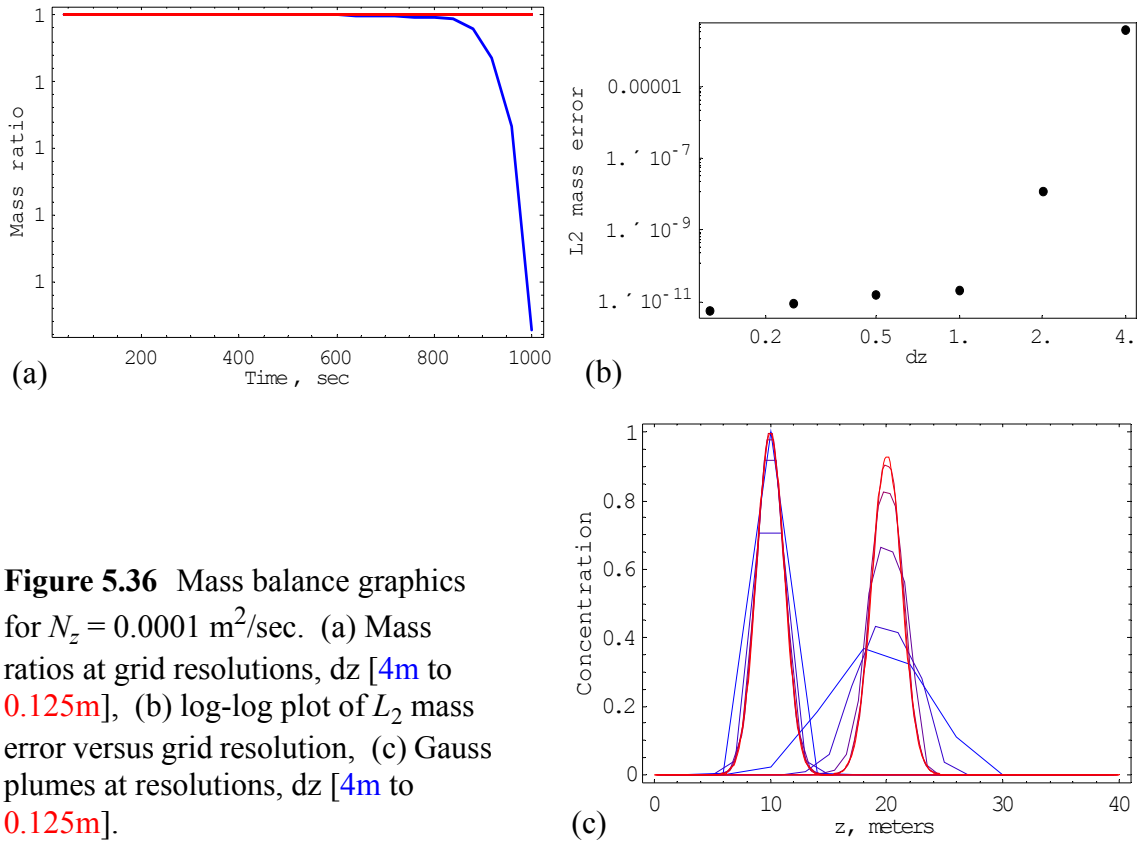
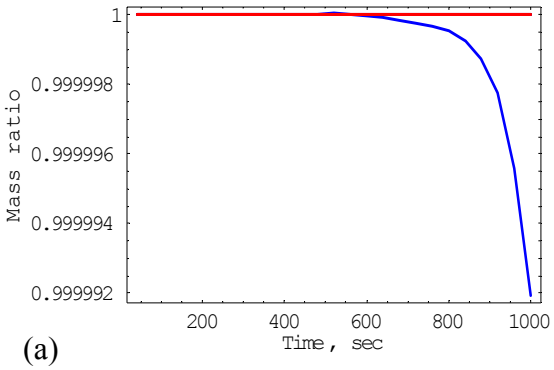


Figure 5.36 Mass balance graphics for $N_z = 0.0001$ m^2/sec. (a) Mass ratios at grid resolutions, dz [4m to 0.125m], (b) log-log plot of L_2 mass error versus grid resolution, (c) Gauss plumes at resolutions, dz [4m to 0.125m].

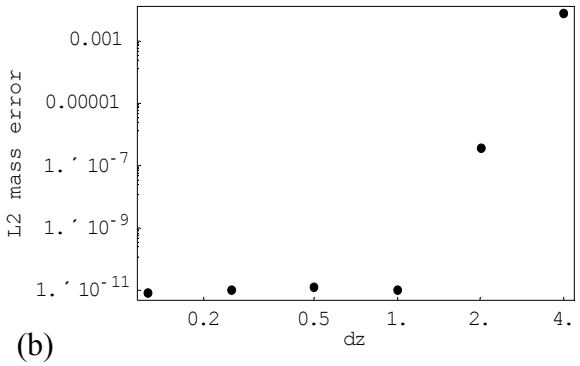
mass balance can be achieved as the grid is resolved. As before, the coarsest resolutions do not capture the peak of the Gaussian distribution, so the initial mass is not equal to exactly 3000 moles; however, the initial mass is conserved. This is evident in the plot of mass ratio and in the convergence results. For grid resolutions less than $\Delta z = 2.0$m, mass balance is near perfect. Notice that the peaks of the final plumes are decreased relative to those for the lower diffusion coefficient in the previous test case.

$N_z = 0.001$ m^2/sec

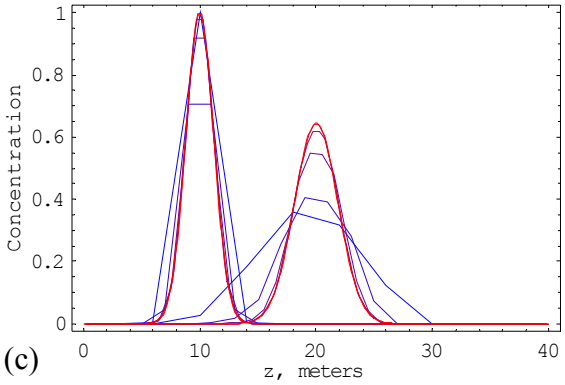
Figure 5.37 shows the mass to initial mass ratios, the L_2 errors, and the initial and final Gauss plumes for a diffusion-dominated test case, and Table 5.18 presents a summary of the error norms. For this diffusion-dominated test case, perfect mass balance can be achieved as the grid is resolved, and all resolutions less than or equal to $\Delta z = 2$m provide



(a)



(b)



(c)

Figure 5.37 Mass balance graphics for $N_z = 0.001$ m^2/sec. (a) Mass ratios at grid resolutions, dz [4m to 0.125m], (b) log-log plot of L_2 mass error versus grid resolution, (c) Gauss plumes at resolutions, dz [4m to 0.125m].

Table 5.18 Spatial errors in mass balance for $N_z = 0.001$ m^2/s.

Δz (m)	Pe	Cr	L_2	L_∞
4	40	0.000025	0.00774373	0.03253454
2	20	0.00005	3.4922E-7	1.4580E-6
1	10	0.0001	1.0331E-11	2.5921E-11
0.5	5	0.0002	1.1889E-11	2.0009E-11
0.25	2.5	0.0004	9.7905E-12	2.1828E-11
0.125	1.25	0.0008	7.9699E-12	1.8190E-11

adequate results. Notice that the peak of the final plumes continue to decrease and the plumes become more dispersed, relative to the previous test cases, as the diffusion coefficient is increased.

$N_z = 0.004\ m^2/sec$

Figure 5.38 shows the mass ratios, the L_2 errors, and the initial and final Gauss plumes for the most diffusive test case. Table 5.19 presents a summary of the error norms. For this most diffusive test case, near perfect mass balance can be achieved as the grid is

Table 5.19 Spatial errors in mass balance for $N_z = 0.004$ m^2/s.

Δz (m)	Pe_z	Cr_z	L_2	L_∞
4	10	0.000025	0.03184391	0.08394053
2	5	0.00005	0.00004997	0.00020058
1	2.5	0.0001	9.9966E-9	4.7490E-8
0.5	1.25	0.0002	3.2383E-9	1.5856E-8
0.25	0.625	0.0004	1.1062E-8	5.4536E-8
0.125	0.3125	0.0008	1.5807E-8	7.7731E-8

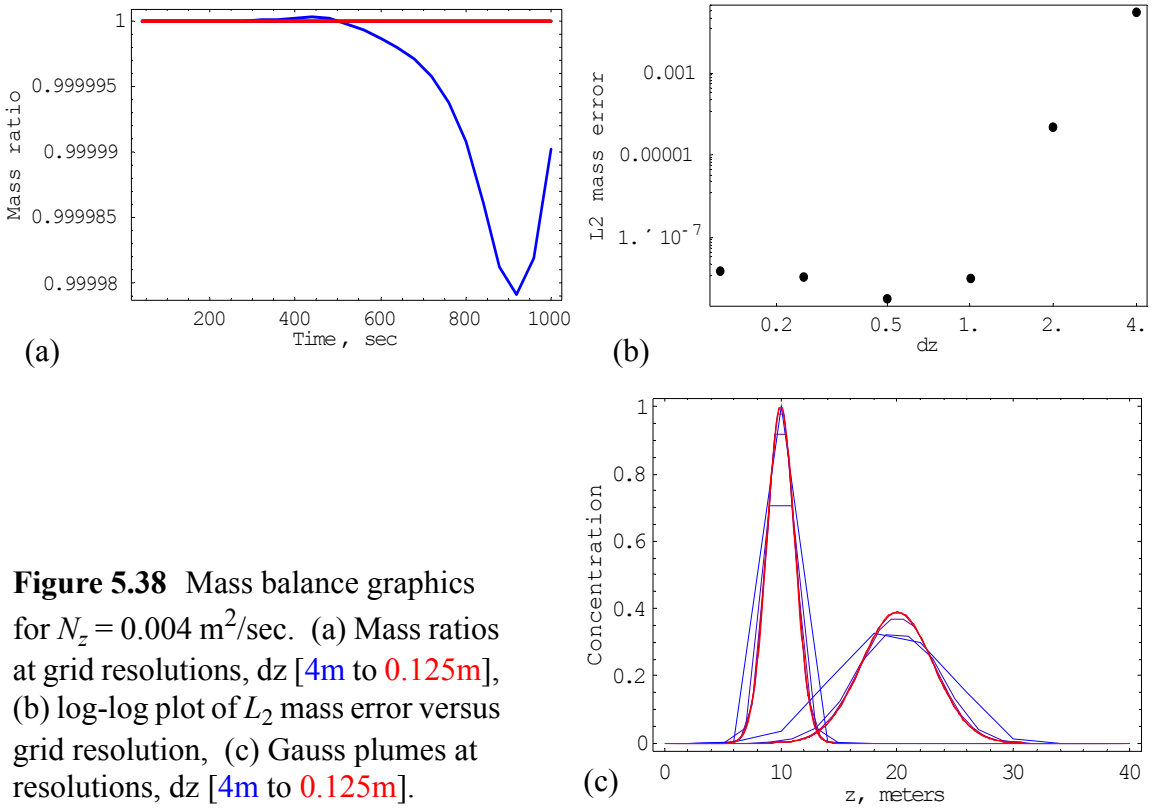


Figure 5.38 Mass balance graphics for $N_z = 0.004$ m^2/sec. (a) Mass ratios at grid resolutions, dz [4m to 0.125m], (b) log-log plot of L_2 mass error versus grid resolution, (c) Gauss plumes at resolutions, dz [4m to 0.125m].

resolved; the error is in the eighth decimal place. Resolutions at or below Δz = 1.0m are required for a good solution, but resolutions up to Δz = 2.0m are also acceptable. Notice that the peaks are decreased and the plumes themselves are more diffused, relative to the lower diffusion coefficients in the previous test cases. As in the *x*-direction mass balance study, with the higher diffusion some of the mass is allowed to escape at the right boundary when Type II BCs are used.

5.7.3 Temporal stability and accuracy for 1D test cases

Stability results

A numerical stability analysis was performed for the 1D analytical verification test cases; simulations were run at increasingly larger time steps until instability was reached. For each value of diffusivity, coarse, intermediate and fine grid resolutions were tested. The accuracy and stability of the algorithm were found to depend upon the Peclet number. In general, for highly diffusive flows (small Pe) the errors were greater and the stability constraint was more restrictive, while for advection-dominated flows (large Pe) the errors were reasonable, given the sharp nature of the front, and the stability constraint was less restrictive. Stability results for the 1D *x*-coordinate verification test cases are presented in Table 5.20 and results for the *z*-coordinate are presented in Table 5.21.

In general, it was found that for Peclet numbers greater than 5, a Courant number less than 0.5 was needed for stability, while for Peclet numbers less than 5, a Courant number less than approximately 0.1*Pe was needed for stability.

Temporal convergence in the *x*-coordinate direction

Temporal convergence tests were done for selected spatial resolutions and diffusivities for both the analytical verification and mass balance test cases. For the *x* coordinate, temporal convergence was done for N_x = 1.0 and 25.0 m^2/s with a spatial resolution of Δx = 10m for the verification test case and Δx = 20m for the mass balance test

Table 5.20 Stability constraints for x coordinate tests.

N_x (m^2/s)	Δx (m)	max Δt (sec)	Pe_x	Cr_x
0.001	200	100	200,000	0.50
0.001	20	10	20,000	0.50
0.001	2.5	1.25	2,500	0.50
1.0	200	100	200	0.50
1.0	20	10	20	0.50
1.0	2.5	0.7	2.5	0.28
10.0	200	100	20	0.50
10.0	20	5.0	2.0	0.25
10.0	2.5	0.1	0.25	0.04
25.0	200	100	8.0	0.50
25.0	20	2.4	0.8	0.12
25.0	2.5	0.025	0.1	0.01

Table 5.21 Stability constraints for z coordinate tests.

N_z (m^2/s)	Δz (m)	max Δt (sec)	Pe_z	Cr_z
0.000001	4	200	40,000	0.50
0.000001	0.5	25	5,000	0.50
0.000001	0.25	12.5	2,500	0.50
0.0001	4	200	400	0.50
0.0001	0.5	25	50	0.50
0.0001	0.25	12.5	25	0.50
0.001	4	200	40	0.50
0.001	0.5	22.5	5	0.45
0.001	0.25	7.0	2.5	0.28
0.004	4	200	10	0.50
0.004	0.5	8.5	1.25	0.17
0.004	0.25	2	0.625	0.08

case. For the slightly diffusive test case of N_x = 1.0 m^2/s, these resolutions give Peclet numbers of 10 and 20, respectively, while for the diffusion-dominated test case, the Peclet

numbers are 0.4 and 0.8, respectively. For the mass balance test cases, temporal convergence studies were done with all three boundary condition types, as was done for the spatial convergence tests for these two values of diffusivity. For each convergence study, only the L_2 error norms are plotted, but both error norms are presented in tabular form.

Figure 5.39 shows the L_2 error norms and Table 5.22 summarizes the convergence

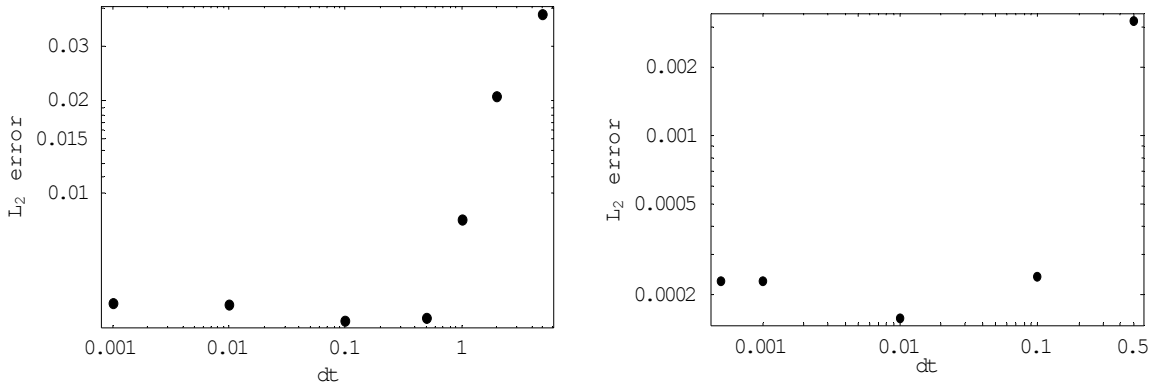


Figure 5.39 L_2 error versus time step for the 1D x verification test cases: (a) $N_x = 1.0$ and (b) $N_x = 25.0$ m^2/sec.

Table 5.22 Temporal convergence for x verification test cases.

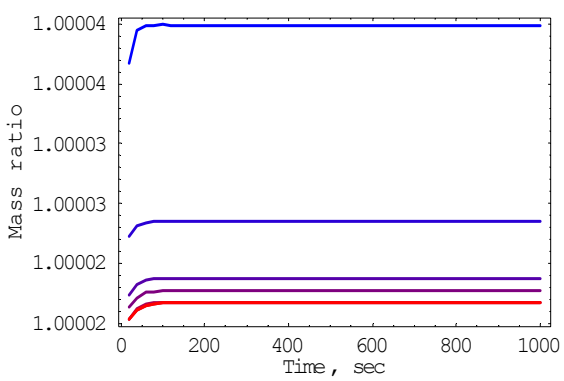
		$N_x = 1.0$ m^2/sec (Pe$_x = 10$)		$N_x = 25.0$ m^2/sec (Pe$_x = 0.4$)	
Δt (sec)	Cr$_x$	L_2	L_∞	L_2	L_∞
5	0.5	0.03826555	0.37479877	unstable	
2	0.2	0.02056503	0.15639614	unstable	
1	0.1	0.00821162	0.05599851	unstable	
0.5	0.05	0.00389862	0.05627594	0.00318000	0.01823136
0.1	0.01	0.00383753	0.05960612	0.00024145	0.00291052
0.01	0.001	0.00430232	0.06029930	0.00015764	0.00233952
0.001	0.0001	0.00435393	0.06036653	0.00022944	0.00228256
0.0005	0.00005	na		0.00022984	0.00227940
Peak rate:		1.324	1.482	1.602	1.140
Average rate:		1.026	1.218	0.893	0.617
Best-fit linear rate:		1.013	1.169	0.728	0.495

results for the verification test cases. The average and best fit rates are computed for the linear portion of the curves before they level off. Notice that for N_x = 1.0 m^2/sec the solutions have converged by Δt = 0.5 seconds, while for N_x = 25.0 m^2/sec the solutions have converged by Δt = 0.1 seconds. Therefore, the time step of 0.001 seconds that was used for the verification test cases is sufficient over the range of diffusion values. However, even at the smallest time step, there will be some error relative to the analytical solution. Also note that all of the convergence rates for both values of diffusivity are near the theoretical value of 1.0 for an explicit time discretization.

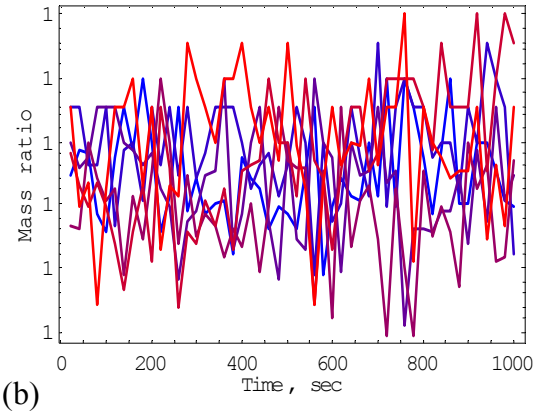
Mass ratio plots at a diffusivity of N_x = 1.0 m^2/sec comparing the boundary types are given in Figure 5.40; convergence results are given in Figure 5.41 and Table 5.23. Note that although the mass ratios with Type II and III BCs appear to be very erratic, the scale of the plots is around the twelfth decimal place, as is evident in the convergence results. For Type I BCs with this low value of diffusivity, the mass builds up to a value slightly higher than the original mass in the domain and then levels out within 100 seconds. Although, this is not a huge error, compared to the other two boundary condition types, mass balance is not achieved with Dirichlet conditions.

Table 5.23 Temporal errors in mass balance for N_x=1.0 m^2/s (Pe$_x$ = 20).

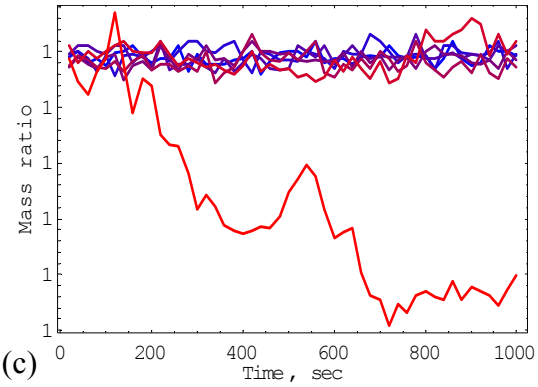
		Type I BCs		*Type II BCs*		*Type III BCs*	
Δt (sec)	Cr$_x$	L_2	L_∞	L_2	L_∞	L_2	L_∞
10	0.5	0.039815	0.039922	1.7637e-12	4.0927e-12	1.5804e-12	3.9790e-12
5	0.25	0.023445	0.023479	1.4633e-12	4.0927e-12	1.5820e-12	4.3201e-12
2	0.1	0.018671	0.018711	2.0623e-12	5.6843e-12	1.7085e-12	3.6380e-12
1	0.05	0.017660	0.017701	2.7454e-12	6.0254e-12	1.8576e-12	5.0022e-12
0.1	0.005	0.016740	0.016782	2.1774e-12	5.1159e-12	2.6682e-12	5.5707e-12
0.01	0.0005	0.016655	0.016697	1.9842e-12	5.0022e-12	2.9646e-12	6.1391e-12
0.001	0.00005	0.016647	0.016689	3.7624e-11	5.5024e-11	3.2056e-11	4.9113e-11



(a)

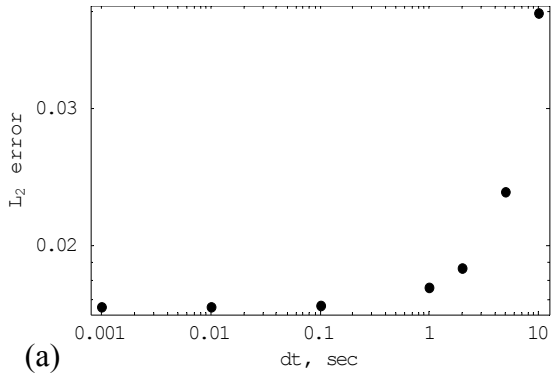


(b)

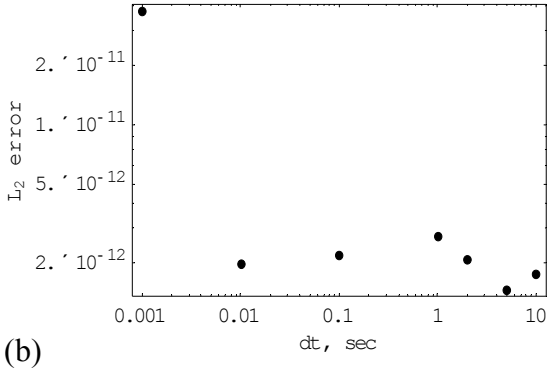


(c)

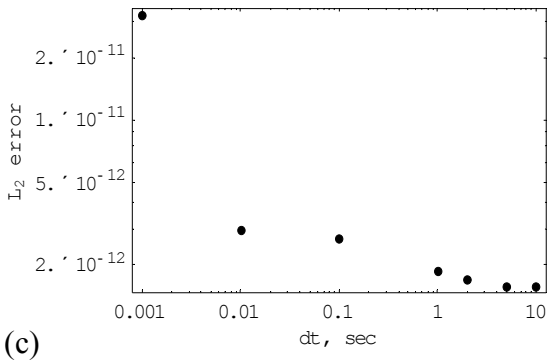
Figure 5.40 Mass ratios versus time for N_x = 1.0 m^2/sec at temporal resolutions dt [0.001 to 10.0 sec]: (a) Type I , (b) Type II and (c) Type III BCs.



(a)



(b)



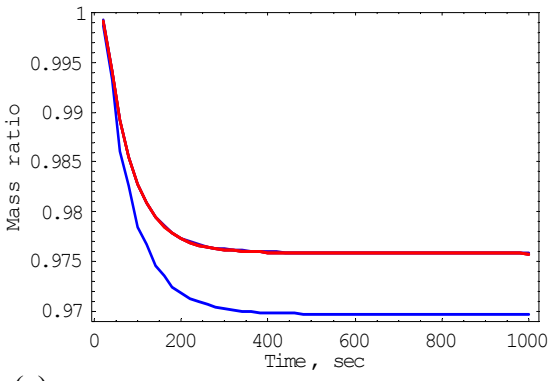
(c)

Figure 5.41 L_2 mass error versus time step for N_x = 1.0 m^2/sec with: (a) Type I, (b) Type II and (c) Type III BCs.

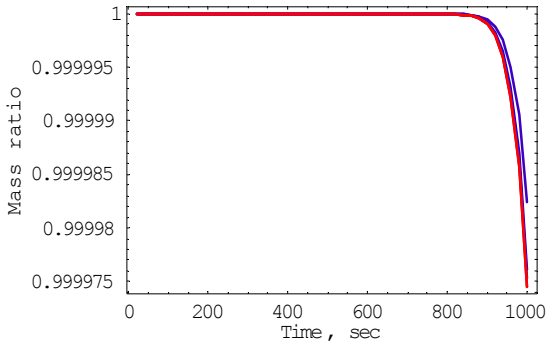
Mass ratio plots at a diffusivity of N_x = 25.0 m^2/sec comparing the boundary types are given in Figure 5.42; convergence results are given in Figure 5.43 and Table 5.24. Note that mass balance with Type I BCs at this high value of diffusion is quite poor; the mass in the domain drops off sharply and levels off after about 500 seconds. By this time, a significant amount of mass has been lost.

Table 5.24 Temporal errors in mass balance for N_x=25.0 m^2/s (Pe$_x$ = 0.8).

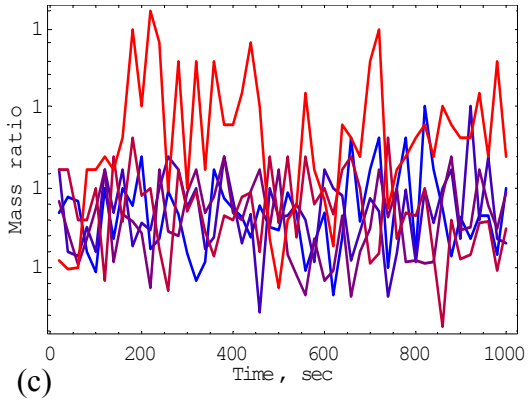
		Type I BCs		*Type II BCs*		*Type III BCs*	
Δt (sec)	Cr$_x$	L_2	L_∞	L_2	L_∞	L_2	L_∞
2	0.1	28.5906	30.3907	0.004009	0.023870	3.0171e-12	6.8212e-12
1	0.05	22.8072	24.1638	0.002926	0.017515	2.8848e-12	7.8444e-12
0.1	0.005	22.8529	24.2308	0.004179	0.024662	3.2255e-12	6.9349e-12
0.01	0.0005	22.8584	24.2385	0.004322	0.025473	3.0215e-12	8.7539e-12
0.001	0.00005	22.8590	24.2393	0.004337	0.025555	5.0720e-11	1.1141e-11



(a)



(b)



(c)

Figure 5.42 Mass ratio versus time for N_x = 25.0 m^2/sec at resolutions dt [0.001 to 2.0 sec]: (a) Type I, (b) Type II and (c) Type III BCs.

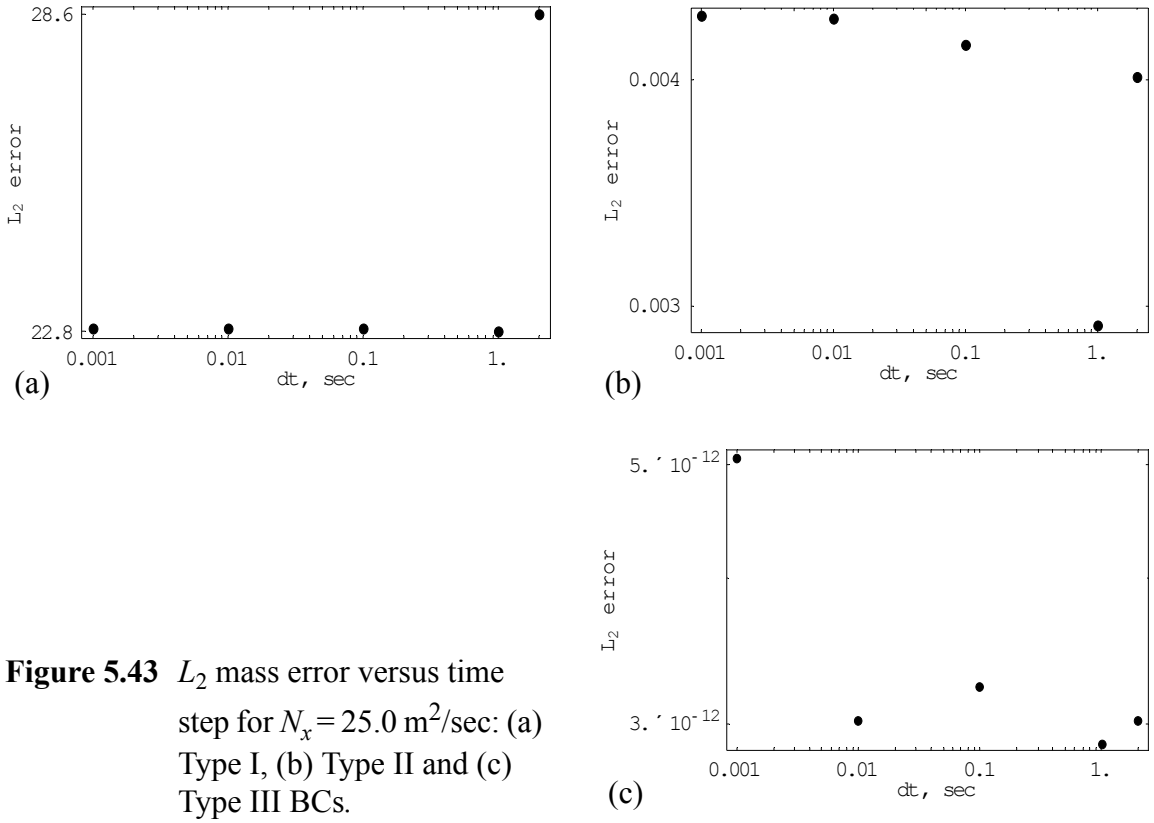


Figure 5.43 L_2 mass error versus time step for $N_x = 25.0$ m^2/sec: (a) Type I, (b) Type II and (c) Type III BCs.

For Type II BCs with high diffusion, the mass begins to decline sharply after about 800 sec. A zoomed in view of Figure 5.42(b) would show that the drop off actually becomes significant somewhat after t=600sec. Due to the highly diffusive nature of this test case, mass is leaving the domain at the right boundary as the plume has dispersed enough that it has reached the boundary at this time. Error analysis of only the first 600 seconds when the entire plume is within the domain, reveals that the errors are of the same magnitude as those for the less diffusive test case with $N_x = 1.0$ m^2/sec. Table 5.25 shows the mass convergence results for this shorter time frame. This is more indicative of the algorithm's capability since the larger time allows mass to escape through the right boundary.

Both the spatial and temporal convergence results indicate that Type I BCs somehow lose mass over the simulations. This is more evident with the higher values of diffusion, which allow the plume to spread further through the domain. This “loss” of mass

Table 5.25 Temporal errors in mass balance for N_x=25 m^2/s at an endtime of 600 seconds using Type II boundary conditions.

		Nx = 25.0 (Pe = 0.8)	
Δt (sec)	Cr	L_2	L_∞
2	0.1	2.67937e-12	6.25278e-12
1	0.05	2.58988e-12	5.34328e-12
0.1	0.005	3.67012e-12	7.27596e-12
0.01	0.0005	3.2179e-12	6.82121e-12
0.001	0.00005	5.21509e-12	1.03455e-11

is due to the fact that Type I BCs create an artificial sink at the boundaries. By forcing the boundary value to be always equal to zero, mass is lost that would have otherwise "piled" up at the boundary as the plume reaches the end of the domain. Thus, the transport algorithm itself is still conservative; although the BCs add a sink term to the solution. In practice, it is best to avoid such artificial boundary conditions, instead using more natural flux-type conditions.

Temporal convergence in the *z*-coordinate direction

Temporal resolution convergence tests were done for select spatial resolutions and diffusivities for both the analytical verification and mass balance test cases in the *z* coordinate. Specifically, temporal convergence was done for N_z = 0.0001 and 0.004 m^2/sec with a spatial resolution of Δz = 0.25m for the verification test case and Δz = 0.5m for the mass balance test case. For the slightly diffusive test case of N_z = 0.0001 m^2/sec, these resolutions give Peclet numbers of 25 and 50, respectively, while for the diffusion-dominated test case, the Peclet numbers are 0.625 and 1.25.

Figure 5.44 shows the L_2 error norm versus time step and Table 5.26 summarizes the convergence results for the *z* verification test cases. The average and best fit rates are computed for the linear portion of the curves before they level off. Note that for both values of diffusivity, a time step of 0.1 sec is sufficient as there is no significant improvement for

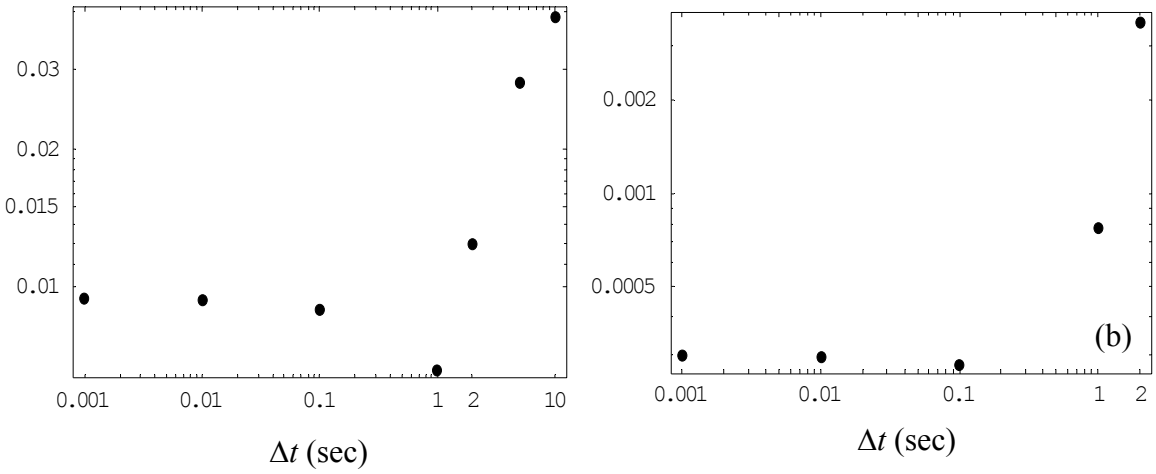


Figure 5.44 Log-log plot of L_2 error versus time step for the 1D z verification test cases: (a) N_z= 0.0001 and (b) N_z = 0.004 m^2/sec.

Table 5.26 Temporal convergence for z verification test cases.

		N_z = 0.0001 m^2/sec (Pe$_z$ = 25)		N_z = 0.004 m^2/sec (Pe$_z$ = 0.625)	
Δt (sec)	Cr$_z$	L_2	L_∞	L_2	L_∞
10	0.4	0.03904122	0.35260673	unstable	
5	0.2	0.02791950	0.25693617	unstable	
2	0.08	0.01243569	0.09858941	0.00359913	0.03280950
1	0.04	0.00660316	0.09381649	0.00077437	0.00884746
0.1	0.004	0.00892983	0.09995581	0.00027988	0.00502378
0.01	0.0004	0.00940659	0.10147721	0.00029712	0.00475637
0.001	0.00004	0.00945514	0.10162741	0.00029982	0.00472988
Peak rate:		0.913	1.045	2.217	1.891
Average rate:		0.760	0.751	1.329	1.068
Best-fit linear rate:		0.787	0.805	0.749	0.531

smaller steps. Also, the convergence rates for both diffusion coefficients are near the theoretical value of unity; the rates for the higher diffusion coefficient are slightly higher than first-order.

Plots of the mass ratios versus time and L_2 mass errors versus time step for the z coordinate mass balance tests are given in Figure 5.45, and Table 5.27 summarizes the convergence results for both values of diffusivity. Note that the scale of the erratic behavior for $N_z = 0.0001$ m^2/sec is 10^{-12} and mass is conserved for any temporal resolution with this problem. The downward trend of the mass ratio for both test cases, but particularly the diffusion-dominated test case, suggests that some mass is being lost through the right boundary. Examination of the simulation results indicate that the plume has reached the right boundary (the boundary values are no longer zero) and mass is escaping the system

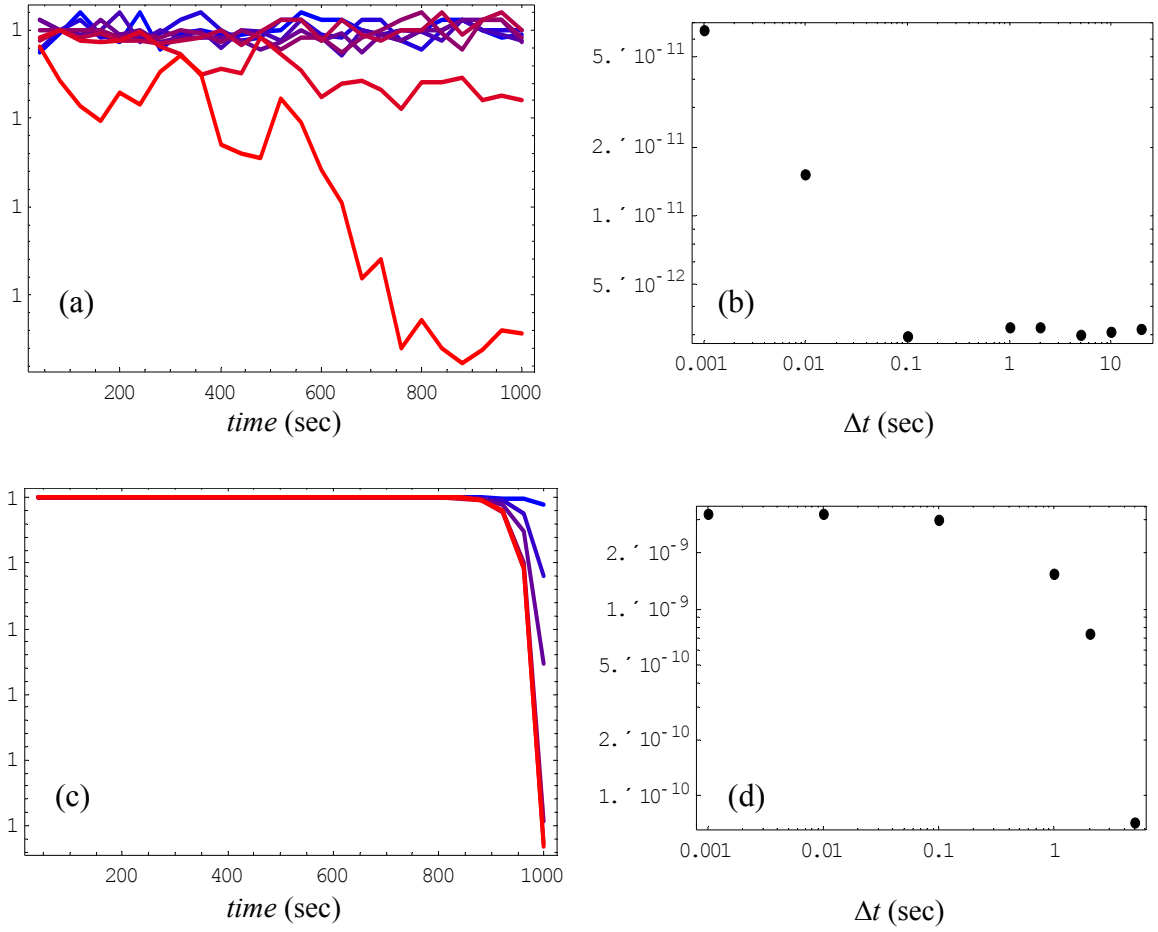


Figure 5.45 Mass balance temporal resolution results for the z-coordinate: (a) and (b) $N_z = 0.0001$ m^2/sec for Δt [0.001 to 20.0 sec], (c) and (d) $N_z = 0.004$ m^2/sec for Δt [0.001 to 5.0 sec].

Table 5.27 Temporal errors in mass balance for z coordinate.

		N_z = 0.0001 m^2/sec (Pe$_z$ = 50)		N_z = 0.004 m^2/sec (Pe$_z$ = 1.25)	
Δt (sec)	Cr$_z$	L_2	L_∞	L_2	L_∞
20	0.4	3.1308e-12	9.0950e-12	unstable	
10	0.2	2.7885e-12	5.9117e-12	unstable	
5	0.1	3.9477e-12	9.5497e-12	5.5562e-6	7.9438e-6
2	0.04	3.2968e-12	9.0950e-12	5.6266e-6	8.0394e-6
1	0.02	3.4465e-12	7.2760e-12	5.6420e-6	8.0631e-6
0.1	0.002	2.6957e-12	9.5497e-12	5.6532e-6	8.0823e-6
0.01	0.0002	6.4640e-12	1.4097e-11	5.6543e-6	8.0843e-6
0.001	0.00002	1.5603e-11	3.0013e-11	5.6544e-6	8.0845e-6

with the Type II BCs. As was seen in the x- direction test case, Type III BCs are the only way to ensure that the system is completely closed. The increase in the mass error as the time step is resolved is indicative of slight differences in the boundary values due to more time steps in the simulation.

When a completely closed system is desired, it would be best to use Type III BCs to enforce a no-flow condition rather than using specified boundary concentrations with Type I BCs. In practice, Type I BCs are used only for algorithm testing since flux conditions (Type II or III) are more natural for actual applications; therefore, for simulations when mass balance is critical, it is advisable to use natural rather than enforced boundary conditions. Furthermore, only Type III BCs are guaranteed to conserve mass since Type II conditions, as specified herein, allow the plume to leave the domain. However, if a small enough grid resolution is used, this can be alleviated. Mass balance with the fully coupled ADCIRC hydrodynamics and LDG transport model is examined in the following section.

5.8 Select results with the x-σ transport model

Although all of the following test cases incorporate a flat-bottom bathymetry and technically can be solved using *x*-*z* coordinates, the ADCIRC hydrodynamics model is solved in the *x*-σ coordinate system. Therefore, in order to communicate between the two models, the transport is also solved in the *x*-σ coordinate system.

5.8.1 Constant velocity tests

For these simple tests, the domain of interest is a 200m by 100m box. The boundary conditions are all set to Type III, such that the box is completely closed. On the interior of the box, the horizontal velocity is set to u = 1.0 m/s, while the vertical velocity is set to w = 0.01 m/s. The grid spacing is equal to Δx = 5.0 m in the horizontal direction and Δz = 4.0 m in the vertical direction, and the diffusivities are set to N_x = 0.5 m^2/sec and N_z = 0.01 m^2/sec. For all simulations, a time step of Δt = 0.1 seconds was used, and the final simulation time was 1000 seconds.

Two different initial conditions are examined: 1) a smooth 2D Gauss plume centered at (x_0 = 40m, z_0 = 25m) and defined as

$$g(x,z) = 35 Exp\left[\frac{(x-x_0)^2}{2\sigma_x^2}\right] Exp\left[\frac{(z-z_0)^2}{2\sigma_z^2}\right], \qquad \textbf{(5.150)}$$

where the Gaussian spreads are σ_x = 20 and σ_z = 10; and 2) a sharp step centered at (x = 30m, z = 37.5m); each of which has a maximum magnitude of 35.0 moles/m^3. These test cases were devised to test the mass balance properties of the LDG transport model in *x*-σ coordinates and to verify that the transport model can handle a sharp gradient between saltwater (35 $^0/_{00}$) and freshwater (typically 4 $^0/_{00}$). The initial conditions are shown in Figure 5.46.

For the smooth Gaussian distribution, the results are shown in Figure 5.47. Notice

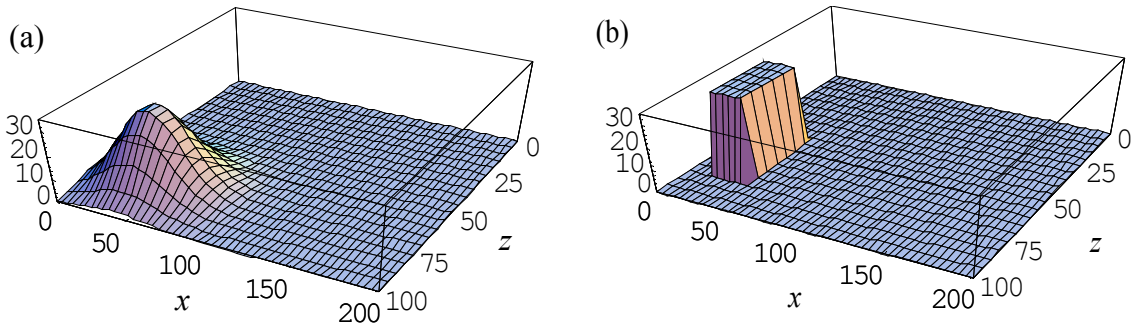


Figure 5.46 Initial salinity profiles for 2D constant velocity test cases: a) Gauss plume and b) sharp step.

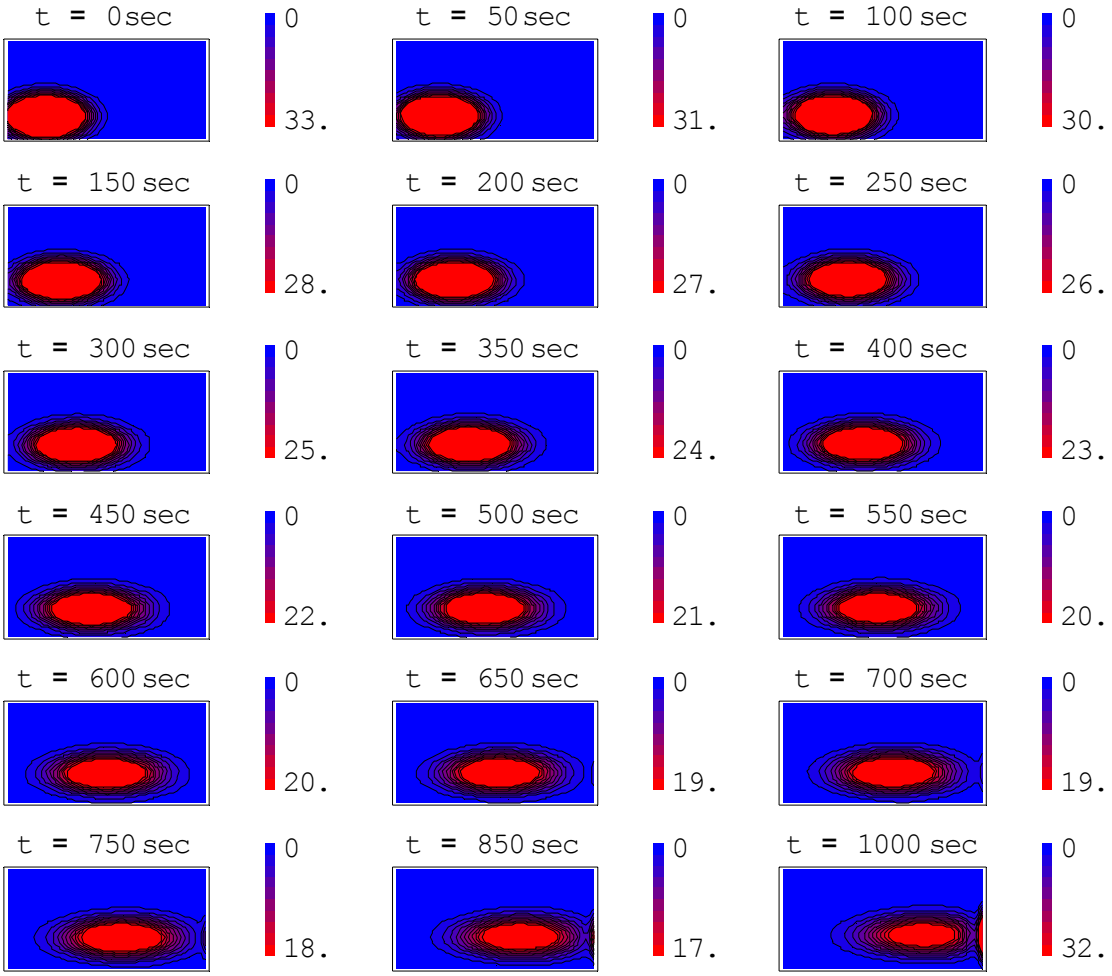


Figure 5.47 Salinity profiles for smooth Gauss plume and constant velocities.

that the plume propagates smoothly without any distortion and slowly diffuses out. As the plume reaches the right boundary, the concentrations begin to "pile" up along the boundary since the domain is closed. A plot of the total mass in the box is shown below in Figure 5.49. Notice that the mass is conserved; the errors are in the twelfth decimal place.

Meanwhile, the simulation results for the sharp step are shown in Figure 5.48. In contrast with the previous example, notice that there are small "wiggles" around the step as it begins to propagate through the domain. However, as the plume is diffused out more, the profile becomes smooth. Although the solution with the sharp IC is not initially smooth, the

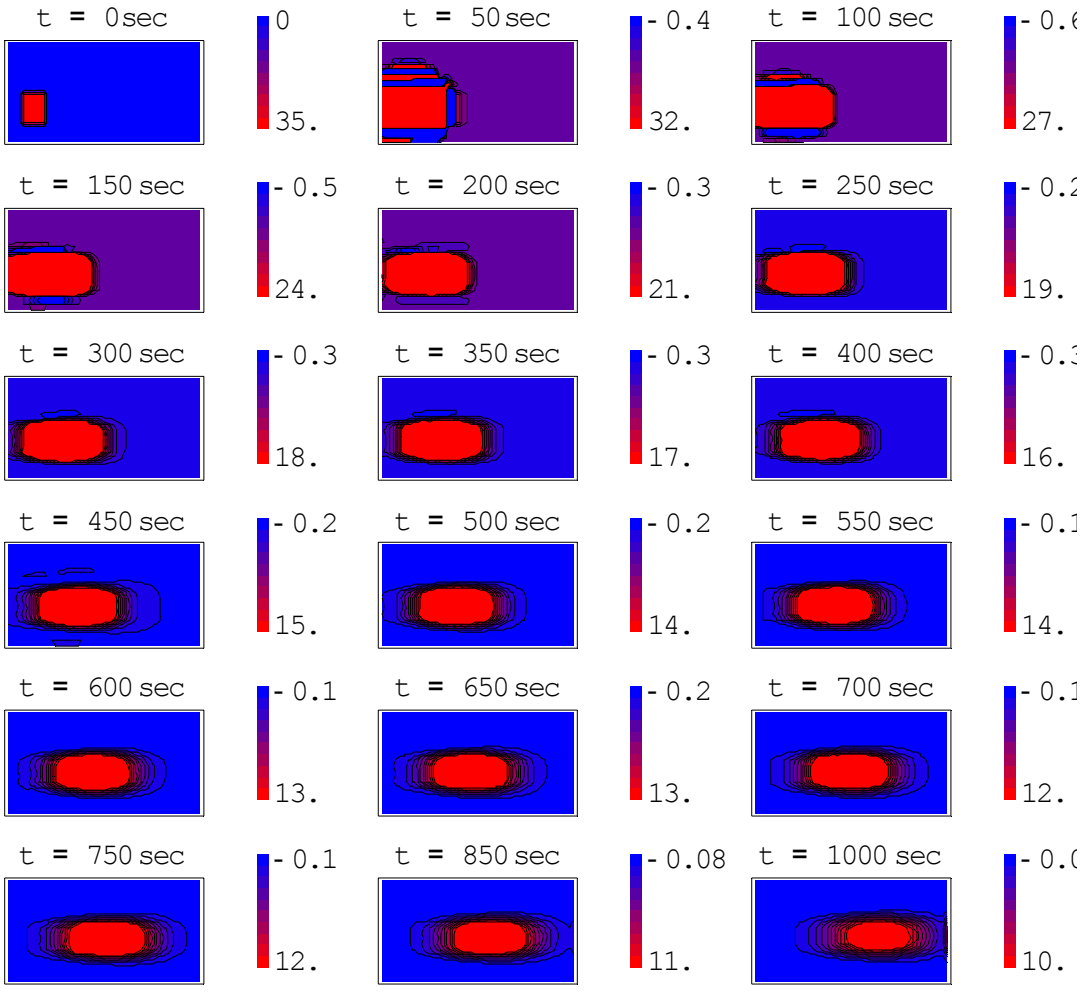


Figure 5.48 Salinity profiles for sharp step initial condition and constant velocities.

total mass in the box is conserved throughout the simulation as is shown in Figure 5.49, where again the errors are in the twelfth decimal place.

These results verify that the LDG transport model itself is mass conservative even when a sharp front is encountered. They also verify that the transport model is capable of handling large gradients without becoming unstable. The next section examines the behavior of the LDG transport model coupled with the ADCIRC hydrodynamic model in x-σ coordinates.

5.8.2 Mixing in a box

This test case consists of an initial salinity gradient in the x coordinate that is constant with depth, a "dam break" of salinity, and was adopted from the tests in *Haidvogel* [1999] in order to examine the mass balance properties and simulation characteristics of the full 2D coupled transport/hydrodynamic model. The simulation domain is a 200m by 100m box. For the hydrodynamics, land boundaries are specified in the x coordinate and kinematic boundary conditions in the σ coordinate; while for the transport model, Type III BCs are specified in both the x and σ coordinates. This combination of boundary conditions results in a completely closed domain, so that mass should be conserved. Furthermore,

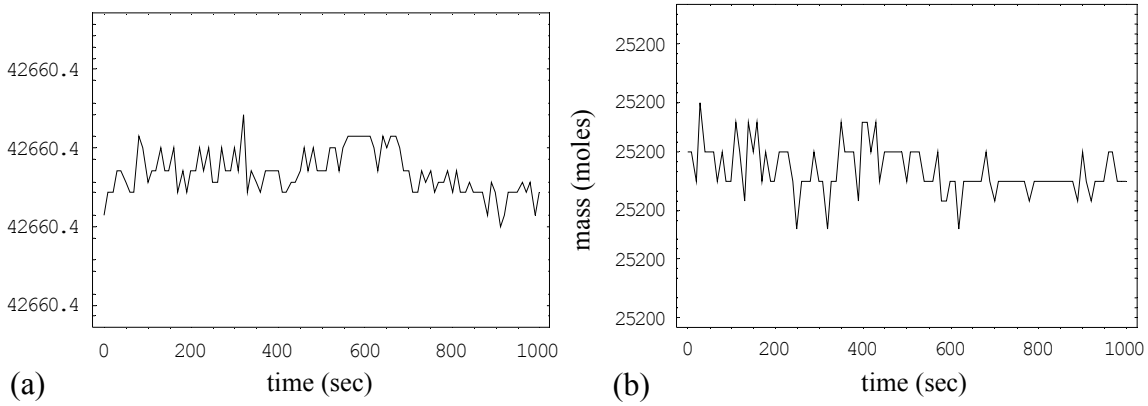
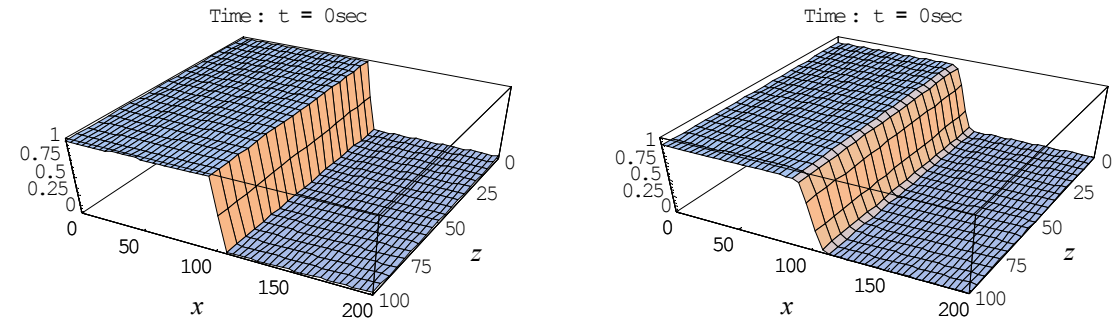


Figure 5.49 Total mass versus time for constant velocity test cases: (a) smooth Gauss plume and (b) sharp step.

since a flat-bottom bathymetric profile is used, the Γ terms of the sigma coordinate transformation are neglected for these test cases. These terms would have to be incorporated for variable bathymetry test cases.

The hydrodynamic model is initialized with zero velocity and elevation everywhere in the domain, such that the system is at rest until the "barrier" between contrasting salinities is removed at time t=0 (see Figure 5.50 below). A total simulation time of 3000 seconds is sufficient to observe the behavior of the system. Two initial gradient profiles are tested: a sharp front, where the change in salinity happens over one element, and a more gradual gradient, where the change occurs linearly over a length of 10m. The magnitude of the gradient was varied to examine the quality of the solution and find the model's limitations. A total of four initial conditions were examined: (1) a gradual gradient from 1.0 to 0.0 – "1grad0", (2) a sharp gradient from 1.0 to 0.0 – "1sharp0", (3) a gradual gradient from 10.0 to 0.0 – "10grad0", and (4) a sharp gradient from 10.0 to 0.0 – "10sharp0". Figure 5.50 shows the initial salinity conditions in the DG grid space with the sharp and gradual gradient profiles when a total gradient of 1.0 is used.

Although there is no analytical solution to verify these test cases, physics dictates that the box will come to a stable solution where the heavier, more dense water will sink to



(a) (b)

Figure 5.50 Initial salinity profiles for mixing in a box test case: (a) sharp front, (b) gradual linear drop.

the bottom, displacing the lighter water, which will rise to the top to replace it. Due to the mixing and diffusion that will occur during this displacement, it is not likely that the steady state will be two layers of constant concentration as in the initialization, but rather that several layers will develop. With non-zero diffusion, as time goes to infinity, the salt concentration will approach some equilibrium profile where the diffusion effects are balanced by the gravity effects. As the simulation domain is closed on all four boundaries, it is expected that the initial mass should be conserved.

Base test case

The set of parameter values that were used for the base test case are taken from another study where the LDG transport model described herein was compared to a continuous Galerkin model (see *Atkinson et al.* [2004]) and are given in Table 5.28. A

Table 5.28 Model parameters for mixing in a box base test case.

nx	51	N_x (m^2/sec)	0.5	Δt (sec)	0.1
nz	21	N_z (m^2/sec)	0.01	*End time* (sec)	3000
Δx (m)	4.0	*evis*	2.0	*substeps*	1
Δz (m)	5.0	*evisvert*	1.5	*skipsteps*	1
gradient	1grad0	*interpolation method*	2	*spatial approximation*	minmod linears

selection of the global salinity profiles from the 3000 second time period are shown below in Figure 5.51. Additionally, a time series of the total mass in the domain is shown in Figure 5.52. Looking first at the salinity profiles, one notices that the front propagates smoothly and maintains the sharp leading edge as it moves toward the right boundary. After it reaches the boundary (around t = 300 seconds) it begins to stratify and "sloshes" back and forth across the box until it reaches "near-equilibrium" at t = 3000 seconds.

Now looking at the mass ratios, one notices that the mass ratio computed from the DG salinity field oscillates about 1.0 during the most turbulent mixing in the first ~800

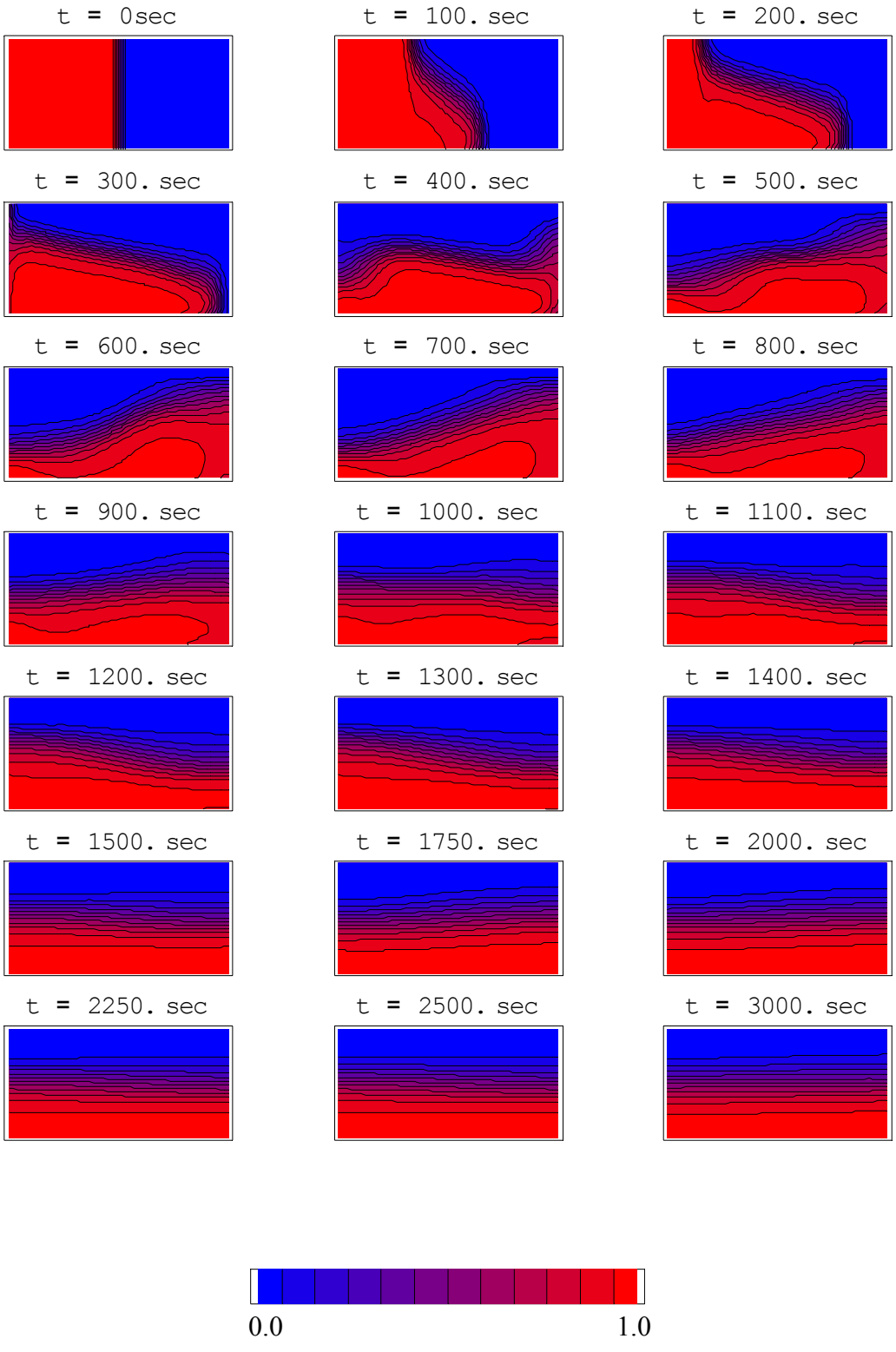


Figure 5.51 Global salinity profiles for “mixing in a box” base test case.

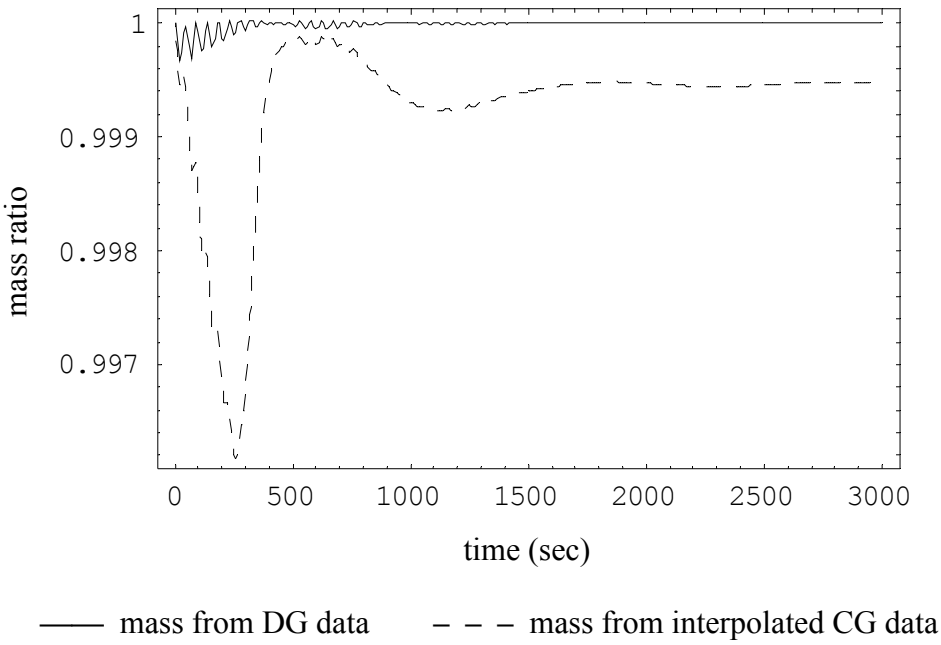


Figure 5.52 Ratio of mass to initial mass versus time for the base test case.

seconds, but then the ratio settles down at unity. Meanwhile, the mass computed from the interpolated CG salinity field (dashed line in Figure 5.52) is significantly lower than that computed from the DG salinity field, which indicates that the interpolation process is not mass conservative. This will be examined further in the parameter studies section. Additionally, note that the coupled transport/hydrodynamic model is no longer perfectly mass conservative despite the conservation properties of the transport algorithm itself. This can be explained by examining the sigma coordinate transformation. Although the Γ terms are neglected for these flat-bottom simulations, the stretching factor, $(a–b)/H$, depends upon the total depth. This depth changes due to the deviations in the sea surface elevation, ζ; and these changes are most significant during the turbulent mixing when the front is propagating across the "box". However, note that the DG mass is still conserved in an "average" sense, as the ratio oscillates about unity even during this turbulent stage.

Stability limits and temporal convergence

For the stability analysis, all of the parameters from the base test case, as given in Table 5.28, remained constant with the exception of the time step. For this test case, the solution remained stable for any time step smaller than 2 seconds. To examine the temporal convergence of the baroclinic model, four time steps were examined, as are summarized in Table 5.29.

Table 5.29 Time steps for temporal convergence and stability.

	base	test1	test2	test3
Δt	0.1	1.0	0.01	0.001

In order to verify that the simulations were converging to the same solution, the 0.5 salinity contour was tracked along the bottom of the domain. This information was used to plot the location of the 0.5 contour versus time for each of the temporal tests. Figure 5.53 shows these contours for the base case and the three tests. Note that the three smallest time

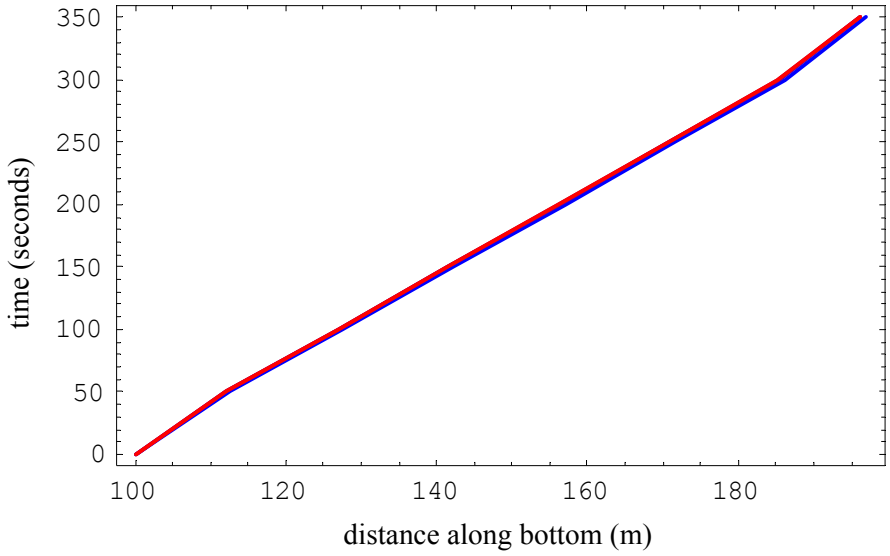


Figure 5.53 Location of 0.5 salinity contour along the bottom of the domain for temporal interplay tests: base (purple), test1 (blue), test2 (maroon), test3 (red).

steps all plot under the red curve, and that the solution with the largest stable time step does not differ significantly. From this metric, it appears that the solutions are converging as the time step is resolved.

Next the mass conservation properties for the four different time steps were examined. Figure 5.54 shows the mass ratio for each of the time steps. Notice that use of the maximum stable time step (Δt = 1.0 seconds) results in the greatest mass oscillations, while use of smaller time steps reduces the mass oscillations. The percent mass errors at the end of the simulation, t = 3000 seconds, for each time step are summarized in Table 5.30. Notice that the mass errors decrease slightly as the time step is reduced, however this reduction in final mass error is not significant.

Table 5.30 Mass errors for temporal convergence.

Δt	1.0	0.1	0.01	0.001
mass error (%)	$6.78E^{-5}$	$6.17E^{-5}$	$5.56E^{-5}$	$5.53E^{-5}$

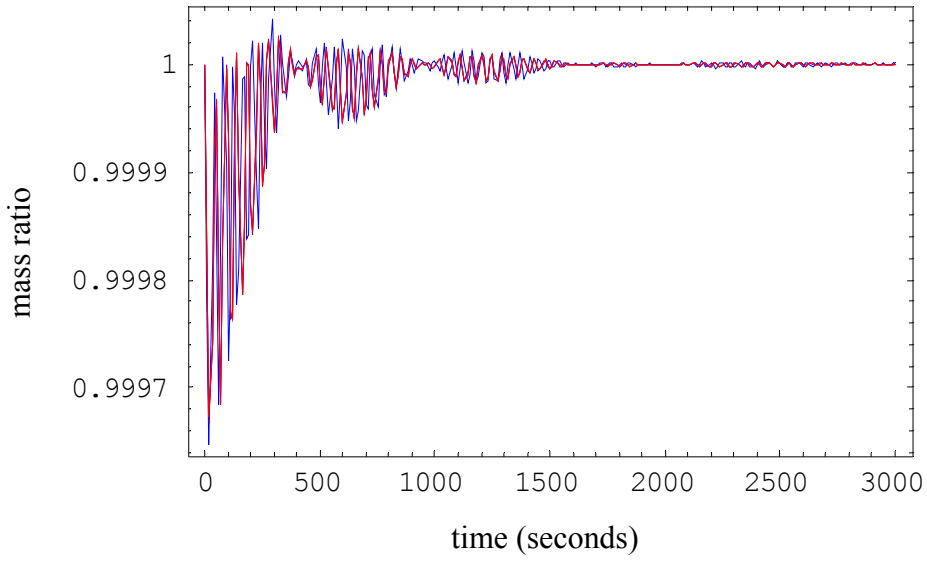


Figure 5.54 Mass ratio for temporal convergence: Δt [1.0 to 0.001] seconds.

Finally, for the maximum stable time step of Δt = 1.0 seconds, the Peclet and Courant numbers, as defined in (5.146) and (5.147), were computed over the entire domain. The global Peclet and Courant numbers for the first output at t = 100 seconds are shown in Figure 5.55. Although the dimensionless numbers are typically reported as an absolute value, they have been graphed with sign in order to show the velocity profile, as well as the magnitude of the dimensionless numbers. The top two panels with the x-coordinate dimensionless numbers, indicate that the horizontal velocity, u, is positive in the bottom half of the box and negative in the upper half of the box. Meanwhile, the bottom two panels indicate that the vertical velocity, w, is positive for the right half of the box and negative for

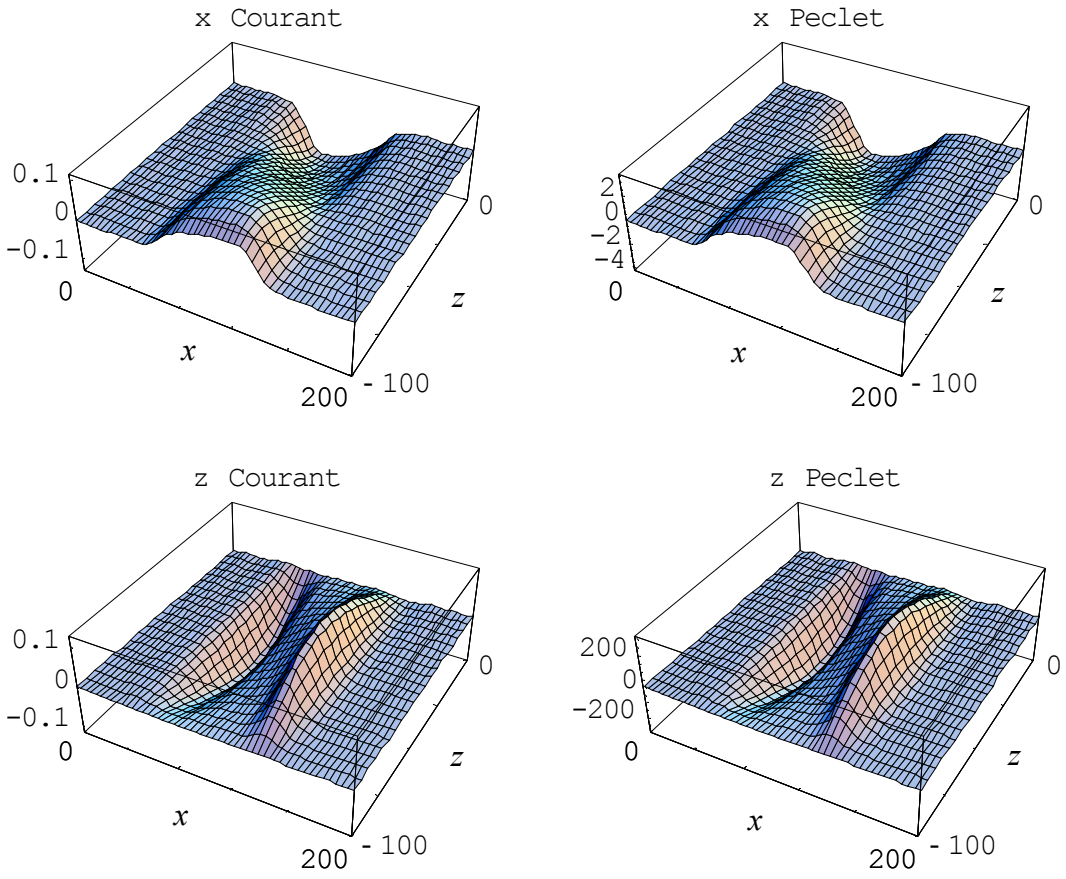


Figure 5.55 Global dimensionless numbers at time, t = 100 seconds, for the maximum stable time step of Δt = 1.0 seconds.

the left half of the box. This is consistent with the flow, where the heavier water is sinking and replacing the light water in the bottom, while the light water is rising and replacing the heavy water from the top.

From a stability point of view, it is easier to look at a plot of the maximum Peclet and Courant numbers over the duration of the simulation. Thus, the minimum and maximum values of these dimensionless numbers are plotted versus time in Figure 5.56. Again, these numbers are shown with sign to illustrate the symmetric nature of the flow; i.e., the positive and negative velocities on opposite sides of the box are of similar magnitude. Notice that the Courant numbers for the *x*- and *z*-coordinate directions are very similar, while the Peclet numbers for the *x*-direction are two orders of magnitude smaller than those in the *z*-direction. Thus the flow in the *x*-coordinate direction is diffusion

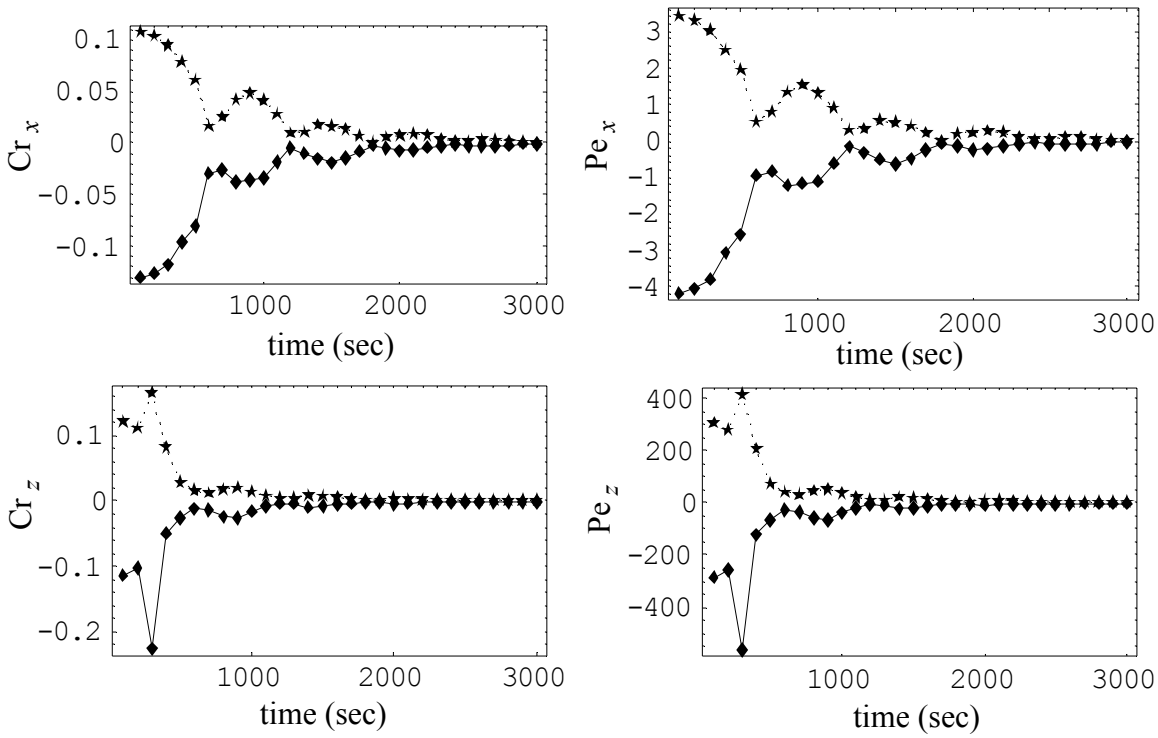


Figure 5.56 Minimum and maximum dimensionless numbers over the domain versus time for the maximum stable time step of $\Delta t = 1.0$ seconds.

dominated, while the flow in the z-direction is more advection dominated, initially. From the definitions, note that the Peclet numbers do not depend on the time step, therefore the Peclet numbers shown in Figure 5.56 do not change for the four different test cases.

From the 1D stability analysis, it was found that for Peclet numbers greater than 5, a Courant number less than 0.5 was needed for stability; while for Peclet numbers less than 5, a Courant number less than approximately 0.1*Pe was needed for stability. From these results, it appears that the flow in the x-coordinate direction is limiting the transport module for this set of parameters. Additionally, the Peclet numbers in both coordinate directions become more indicative of diffusion-dominated flow as the simulation progresses, particularly after the most turbulent mixing ends around t = 1000 seconds. However, it is difficult to determine whether the stability requirements of the full baroclinic ADCIRC/ LDG transport combined model will be most limited by the transport, the hydrodynamics, or due to the complicated interplay between the models; and further testing with more realistic gradients and bathymetries is recommended.

Spatial convergence

For the spatial convergence studies, the set of base parameters given in Table 5.28 was used, with the exception of the grid spacings in the horizontal and vertical directions, Δx and Δz. Three levels of refinement were examined: (1) refine Δx and leave Δz constant at 5.0m, (2) refine Δz and leave Δx constant at 4.0m, and (3) refine both spatial dimensions as a unit. The grid resolutions are summarized in Table 5.31. For each of the three series of refinement, the coarsest resolutions are plotted in shades of blue and the finest resolutions are shown in shades of red.

Table 5.31 Grid resolution combinations for spatial convergence studies.

Refine x coordinate (Δz = 5.0m)	Δx (m)	20	10	4	2	1
Refine z coordinate (Δx = 4.0m)	Δz (m)	20	10	5	2.5	1.25
Refine both coordinate directions	$\Delta x/\Delta z$ (m)	20/20	10/10	4/5	2/2.5	1/1.25

Plots of the mass ratio computed from the DG salinity field and the location of the 0.5 contour, as the x coordinate is resolved, are shown in Figure 5.57. Notice that the finer resolutions have smaller oscillations in the mass ratio as time progresses; but that all of the resolutions conserve mass in an "average" sense. (As discussed for the base test case above.) Note also, that the location of the 0.5 contour along the bottom of the box converges as the grid is refined.

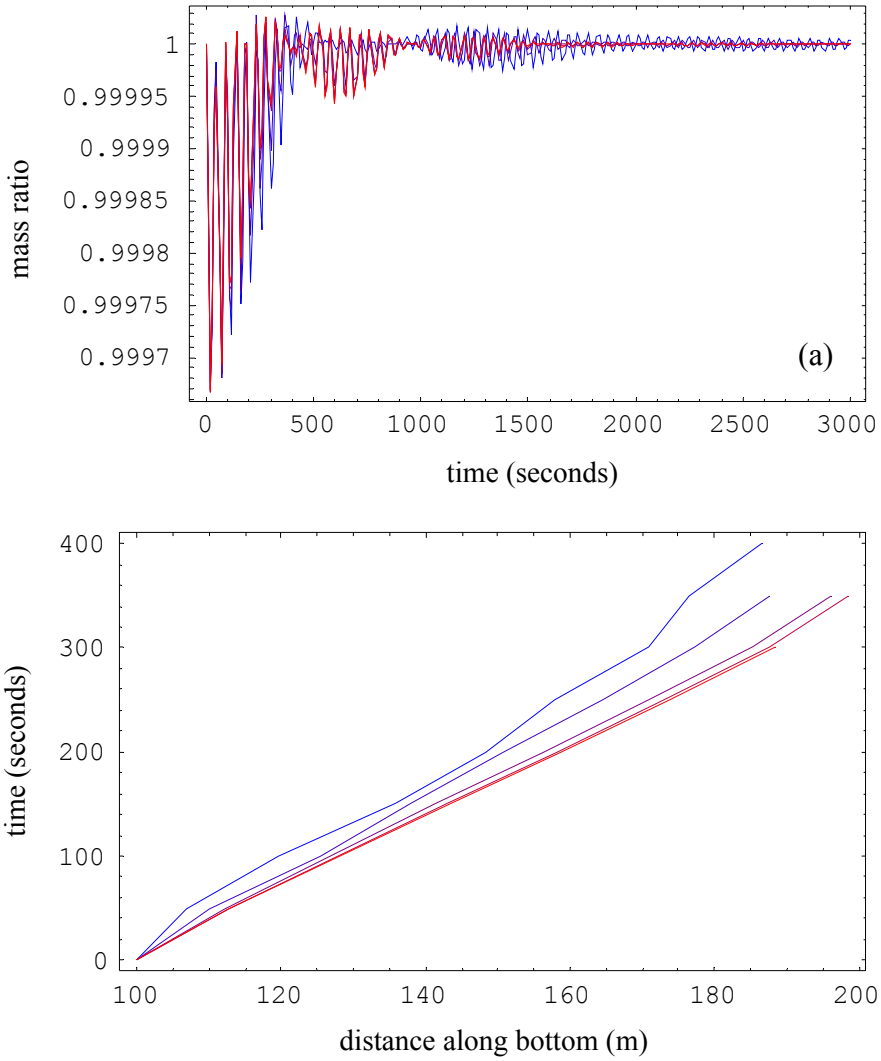


Figure 5.57 Results from the x-coordinate spatial convergence study: (a) mass ratios computed from DG salinity field, (b) location of 0.5 contour along bottom of box. Grid resolutions, Δx [20m – 1m].

Plots of the mass ratio computed from the DG salinity field and the location of the 0.5 contour, as the z coordinate is resolved, are shown in Figure 5.58. Notice that the finer resolutions have smaller oscillations in the mass ratio as time progresses; but that all of the resolutions conserve mass in an "average" sense. Note also, that the location of the 0.5 contour along the bottom of the box converges as the grid is refined. Although the differences for each resolution are not as significant, as those seen for the x-coordinate

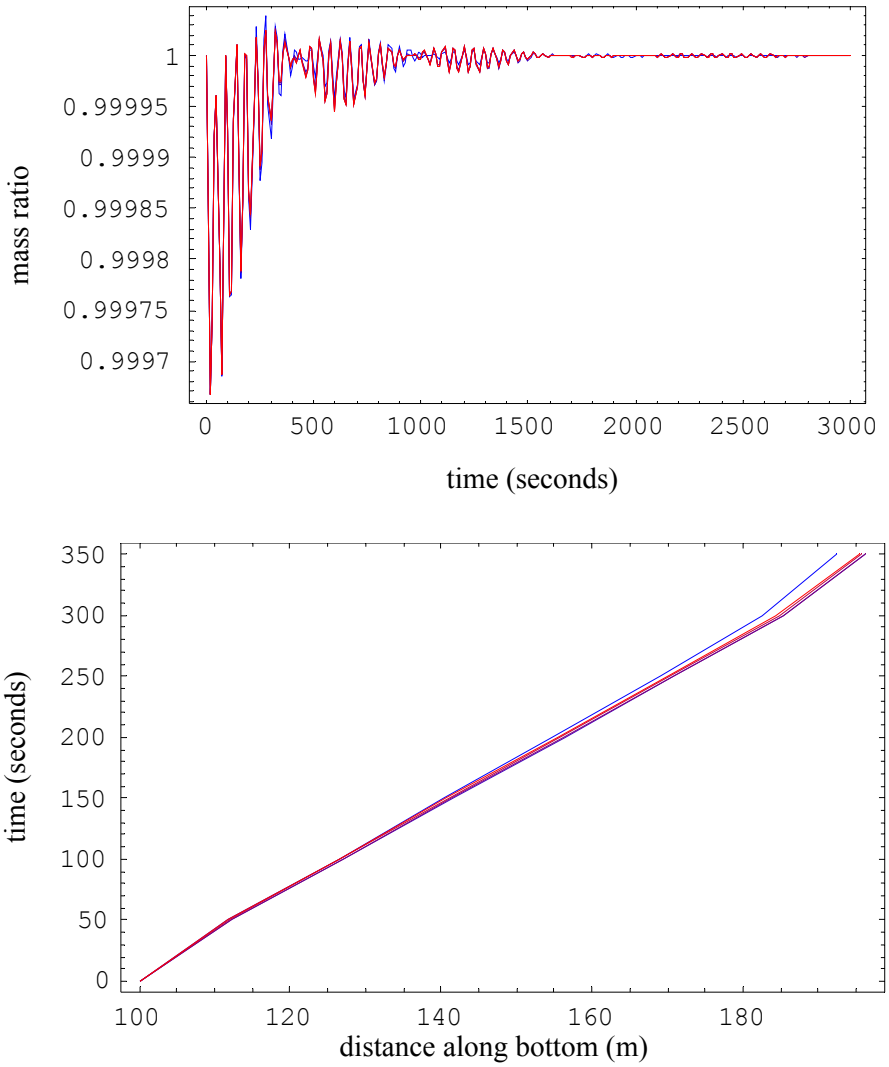


Figure 5.58 Results from the z-coordinate spatial convergence study: (a) mass ratios computed from DG salinity field, (b) location of 0.5 contour along bottom of box. Grid resolutions, Δz [20m – 1.25m].

study, which is to be expected since the x direction is unchanged by the z refinement.

Finally, plots of the mass ratio computed from both the DG and interpolated CG salinity fields and the location of the 0.5 contour, as both coordinate directions are resolved, are shown in Figure 5.59. Notice that the scale of the DG oscillations are too fine to appear in panel (a), although they are similar to those shown in the preceding figures. Also, notice

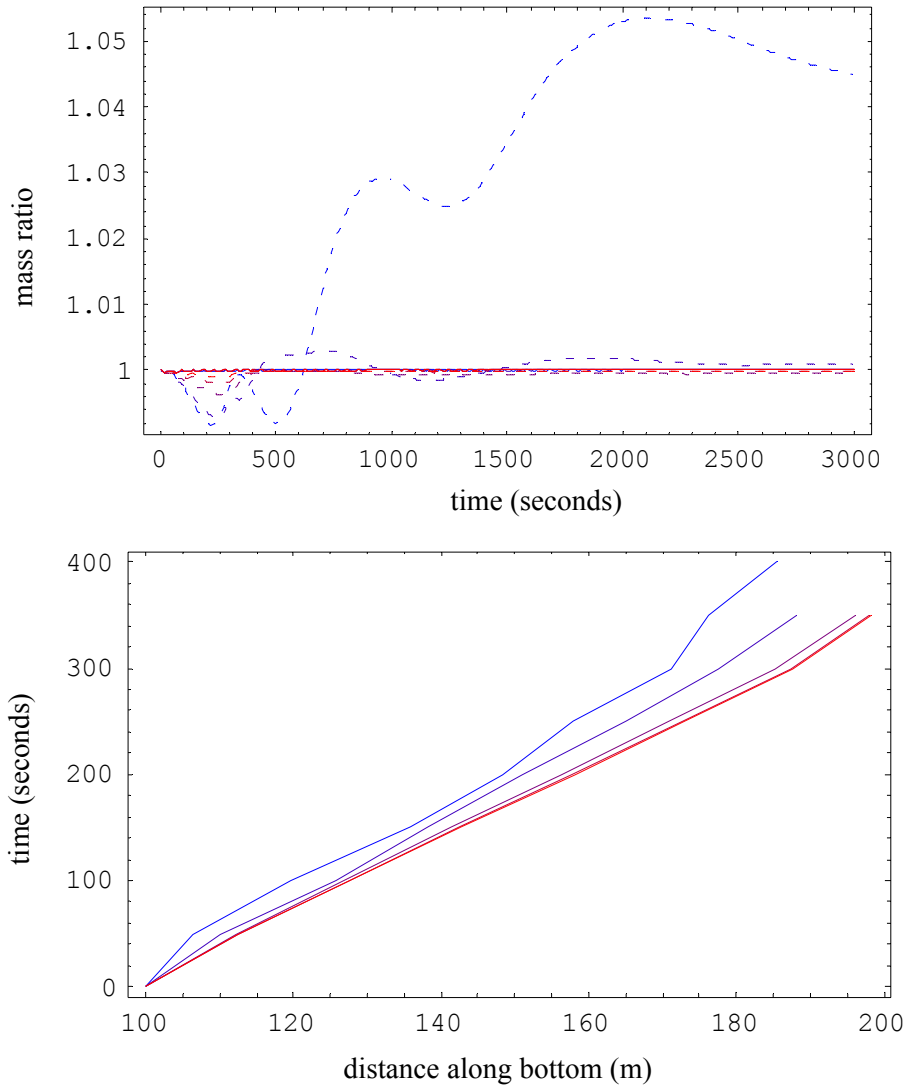


Figure 5.59 Results from spatial convergence in both directions: (a) mass ratios computed from DG (solid) and interpolated CG (dashed) salinity fields, (b) location of 0.5 contour along bottom of box. Grid resolutions, Δx [20m – 1m] and Δz [20m – 1.25m].

that the mass computed from the interpolated CG salinity field improves significantly as the grid is refined. And the location of the 0.5 contour along the bottom of the box converges as the grid is refined and is dominated by the *x*-coordinate grid resolution, as expected.

Parameter studies

A series of parameter studies was also conducted for the lateral and vertical diffusion coefficients and eddy viscosities, the DG to CG interpolation methods, the spatial approximations, and the temporal feedback (*substeps* and *skipsteps*) for each of the gradients in order to determine the role of these parameters on mass balance. The results from the base test case examined above will be included in the plots for each of the parameter studies.

For each of the studies, the mass ratio versus time is plotted over the range of parameter values. The results for the lower parameter values are plotted in shades of blue, with a gradual transition in colors to the higher values in red. The results from the base test case are shown in black. The line thickness and dashing patterns correlate with the gradient type (thick – gradual, thin – sharp) and whether the mass results are from the computed DG (solid) or interpolated CG (dashed) salinities. To keep the problem tractable, the relative magnitudes of the lateral and vertical diffusion and viscosity parameters were kept constant, such that both parameters were varied together as a unit. Four such combinations were examined for the diffusion coefficients, where N_x and N_z are as given in Table 5.32 and the remaining parameters are as in Table 5.28.

Table 5.32 Diffusion coefficient combinations for parameter studies.

N_x (m^2/sec)	0.01	N_x (m^2/sec)	1.0
N_z (m^2/sec)	0.001	N_z (m^2/sec)	0.1
N_x (m^2/sec)	5.0	N_x (m^2/sec)	10.0
N_z (m^2/sec)	0.5	N_z (m^2/sec)	1.0

The mass ratios for the diffusion coefficient parameter study are shown in Figure 5.60. Note that the scales on the two panels are not identical, but that they are similar. In general, as the diffusion coefficients increase, the mass error decreases for all gradient strengths and types. This trend is most significant for the larger gradients. Note also, that as the diffusion coefficients increase, the DG and interpolated CG mass ratios converge to unity as time progresses.

Notice that for the smaller gradient (Figure 5.60a) there is not a discernible difference between the DG results for the two types of gradient, gradual versus sharp, over the entire range of diffusion values. However, the mass ratios computed from the interpolated CG concentrations are slightly different for the two gradient types at the smaller diffusion combinations, with the sharp gradient having slightly better mass properties, but begin to converge as the diffusion increases. Meanwhile, for the larger gradient (Figure 5.60b), the DG results are still not noticeably different for the two gradient types, while the interpolated CG results follow the same trend as for the smaller gradient.

Three additional combinations were examined for the eddy viscosities, where *evis* and *evisvert* are as given in Table 5.33, and the remaining parameters are as in Table 5.28. The mass ratios for these combinations are shown in Figure 5.61.

Table 5.33 Eddy viscosity combinations for parameter studies.

evis	5.0	*evis*	10.0	*evis*	20.0
evisvert	3.75	*evisvert*	7.5	*evisvert*	15.0

In general, as the eddy viscosities are increased, the mass error decreases for the DG mass ratios and increases for the CG mass ratios. The actual error change is not that significant for the smaller gradients, as the final mass ratios differ by only 0.0002 when the eddy viscosities are ten times larger than the base values. Meanwhile, for the larger gradients, the error increase for the CG results is more significant. Note that the final mass

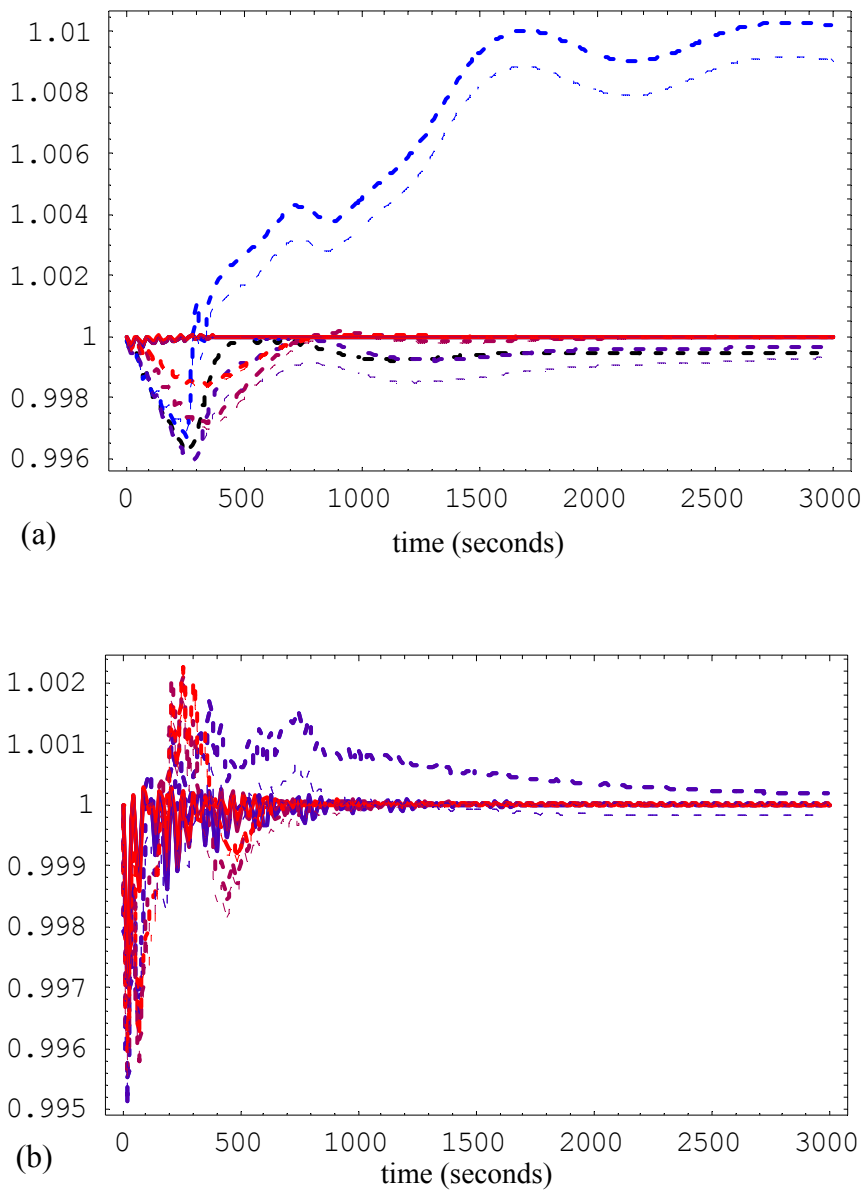


Figure 5.60 Mass ratios versus time for diffusion coefficient parameter study: (a) gradient of 1.0, (b) gradient of 10.0. Diffusion combinations: base (N_x = 0.5 m^2/sec / N_z = 0.01 m^2/sec), 0.01/0.001 (blue), 1.0/0.1 (purple), 5.0/0.5 (magenta), 10.0/1.0 (red).

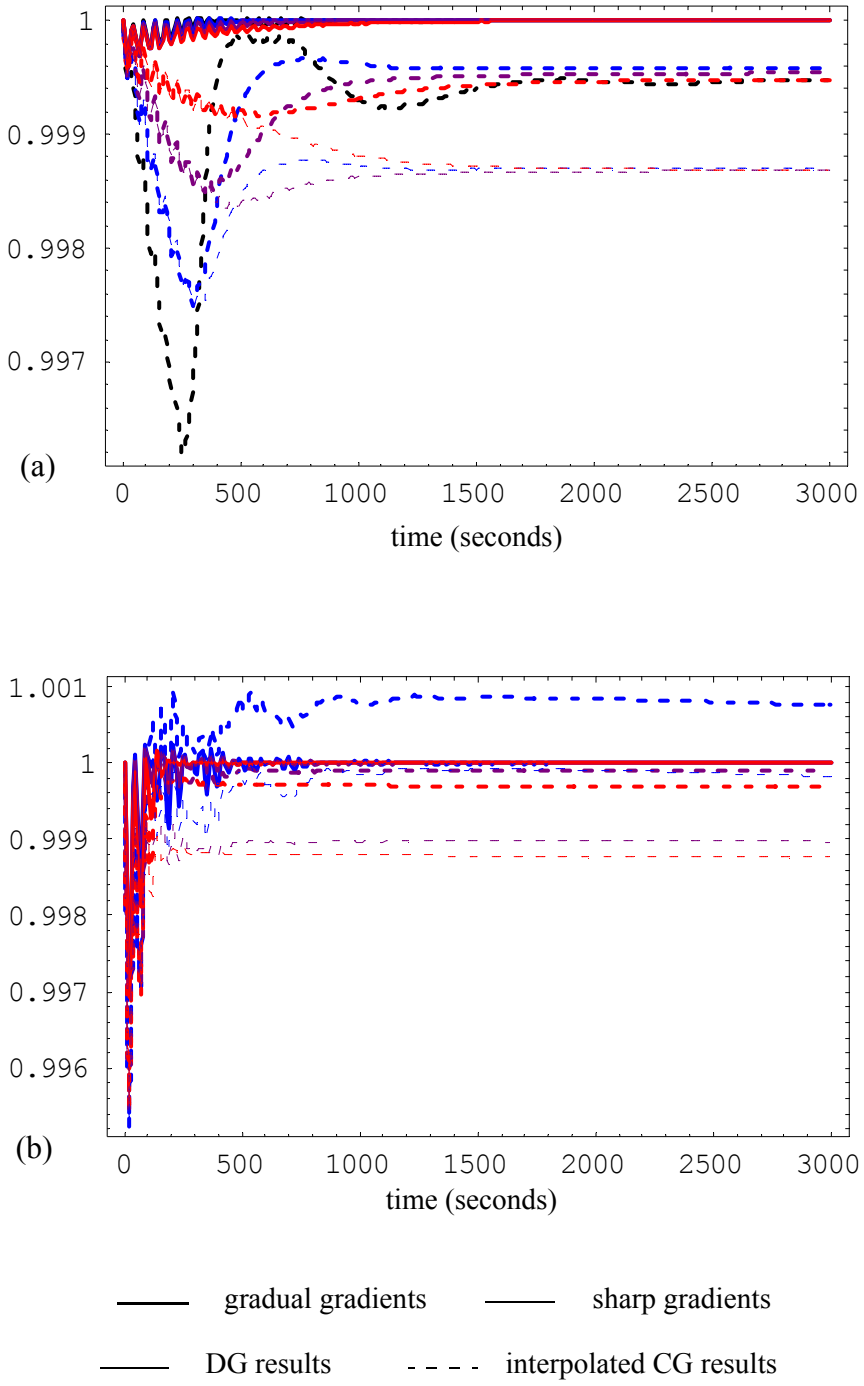


Figure 5.61 Mass ratios versus time for eddy viscosity parameter study: (a) gradient of 1.0, (b) gradient of 10.0. Parameter combinations: base (*evis* = 2.0 / *evisvert* = 1.5), 5.0/3.75 (blue), 10.0/7.5 (purple), 20.0/15.0 (red).

ratios from the interpolated CG results are closer for the small gradient, while they do not converge at all for the larger gradient. Note that unlike the previous graph for the diffusion coefficients, the mass ratios computed from the interpolated CG results do not converge to those computed from the DG results as the eddy viscosity parameters are increased. Notice also that for both gradient strengths, the DG results do not differ for either the gradual or sharp gradient. However, the mass ratios computed from the interpolated CG concentrations are significantly lower for the sharp gradient, when compared to the gradual gradient.

The mass balance behavior of the two interpolation methods are also compared for both gradient strengths. For these tests, the methods for the DG to CG interpolations are numbered as follows: 1) L_2 interpolation over the entire domain and 2) interface averages as computed for the DG solution. The other parameters are as in Table 5.28; while the interpolation method is varied between 1 and 2. The mass ratios for the DG to CG interpolation parameter study are shown in Figure 5.62.

Both interpolation methods have comparable mass behavior for the DG results, as well as for the interpolated CG results for the small gradient. However, there are noticeable differences in the interpolated CG mass results for the higher gradient strength, with the interface average (Method 2) results being better. Notice also that the mass ratios computed from the interpolated CG concentrations have the same general shape for both the sharp and gradual gradients, however, the sharp gradients result in somewhat reduced mass. Again, the mass ratios from both gradient strengths, which are computed from the interpolated CG concentrations, are not conservative. The interface averaging technique was chosen for the base test case, since it has slightly better mass properties than the L_2 interpolation method. However, further work could be done to examine the mass conservation properties of other interpolation methods.

The mass balance behavior of the three spatial approximations, piecewise

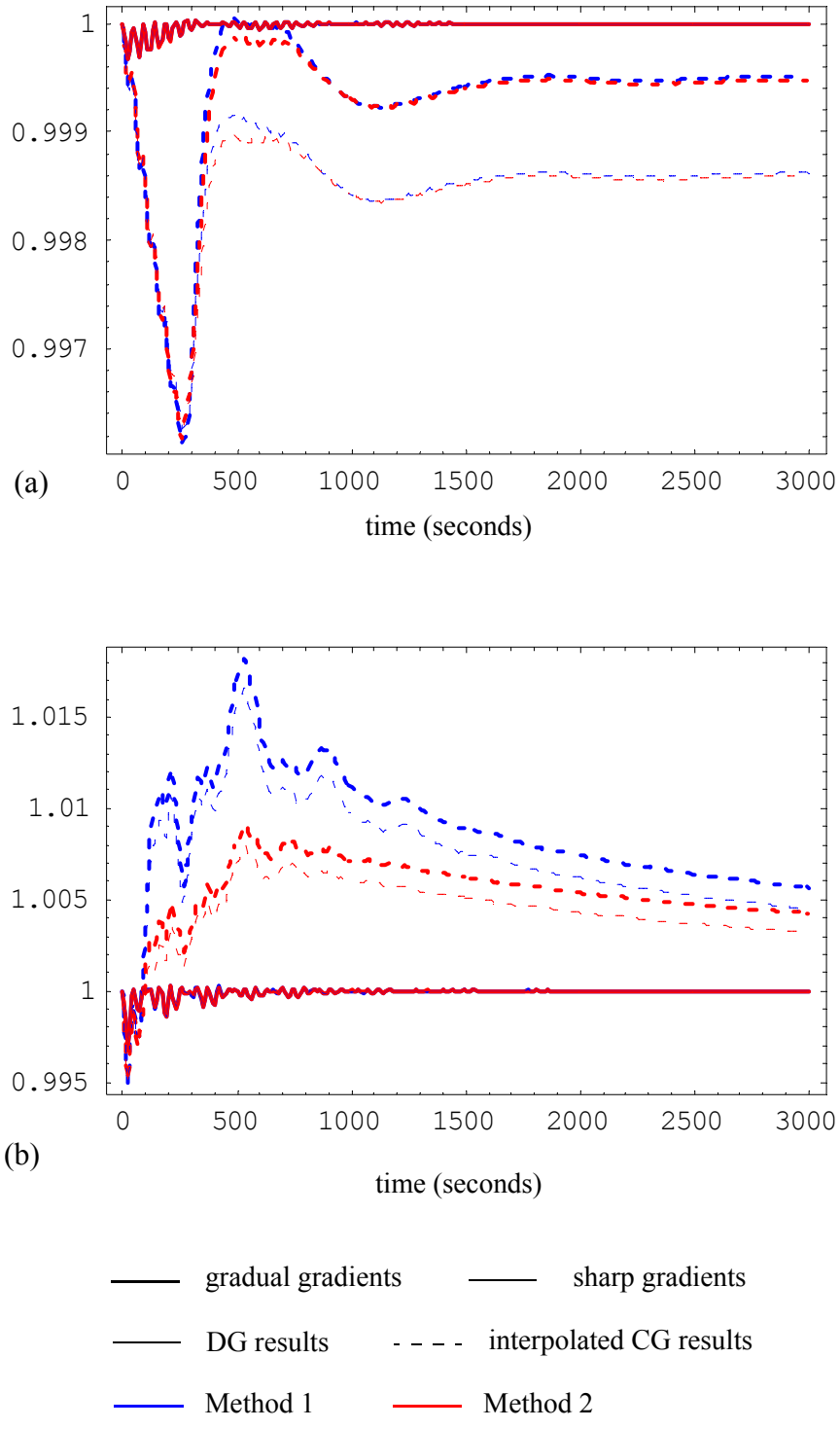


Figure 5.62 Mass ratios versus time for interpolation parameter study: (a) gradient of 1.0 and (b) gradient of 10.0.

constants, unlimited linears, and minmod linears, are also compared for the smaller gradients. The other parameters are as in Table 5.28; while the spatial approximation is varied. The mass ratios for the spatial approximation parameter study are shown in Figure 5.63. Note that both of the higher-order approximations plot under the red curves and that the piecewise constant approximation has slightly better mass properties for the interpolated CG results.

Finally, the role of time scales for the coupling between the transport algorithm and the hydrodynamic model was explored by using two temporal scale parameters: *substeps*, which is the number of transport time steps for each hydrodynamic time step, and *skipsteps*, which is the number of hydrodynamic time steps that should be run before the transport model is updated again. Thus, *substeps* results in a smaller time scale for the transport

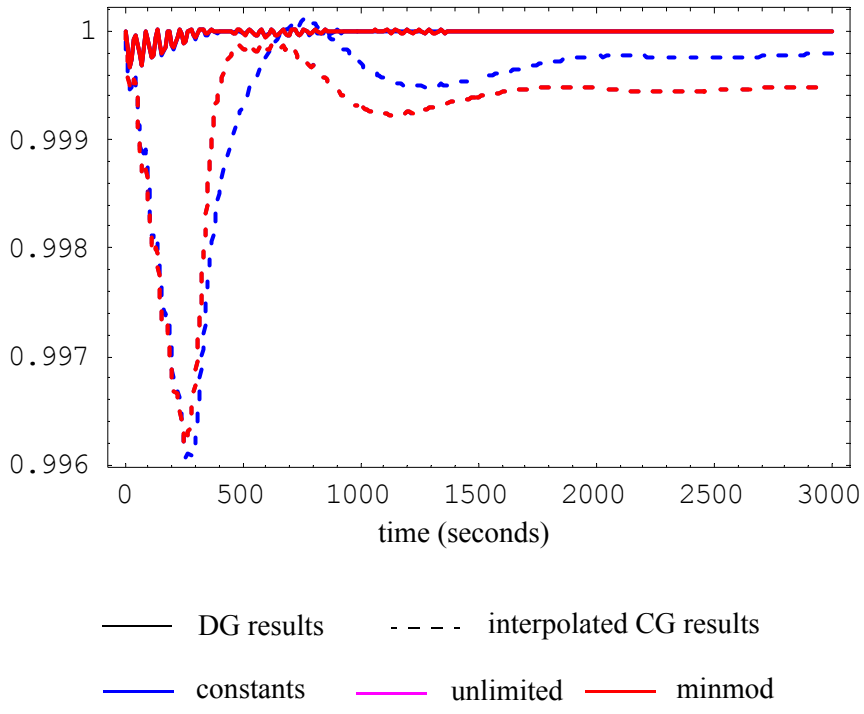


Figure 5.63 Mass ratios versus time for spatial approximation parameter study with a gradient of 1.0.

routine relative to the hydrodynamics and *skipsteps* results in a larger time scale. For the temporal feedback, the time step remained constant at $\Delta t = 0.1$ seconds while the number of times that the transport subroutine was updated varied. Table 5.34 summarizes the combinations of the two temporal parameters, *substeps* and *skipsteps* and the mass ratios for these are shown in Figure 5.64.

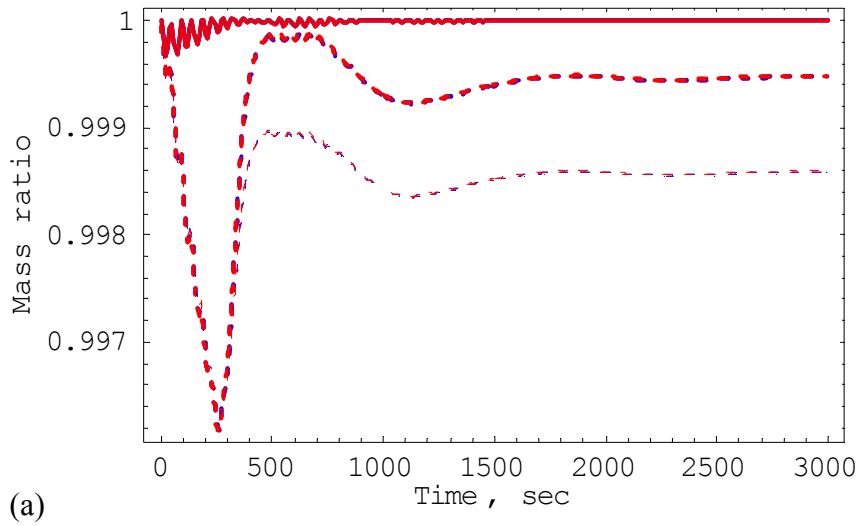
Table 5.34 Temporal combinations for parameter studies.

substeps	5	*substeps*	2	*substeps*	1	*substeps*	1
skipsteps	1	*skipsteps*	1	*skipsteps*	2	*skipsteps*	5
Δt_{eff} (sec)	0.02	Δt_{eff} (sec)	0.05	Δt_{eff} (sec)	0.2	Δt_{eff} (sec)	0.5

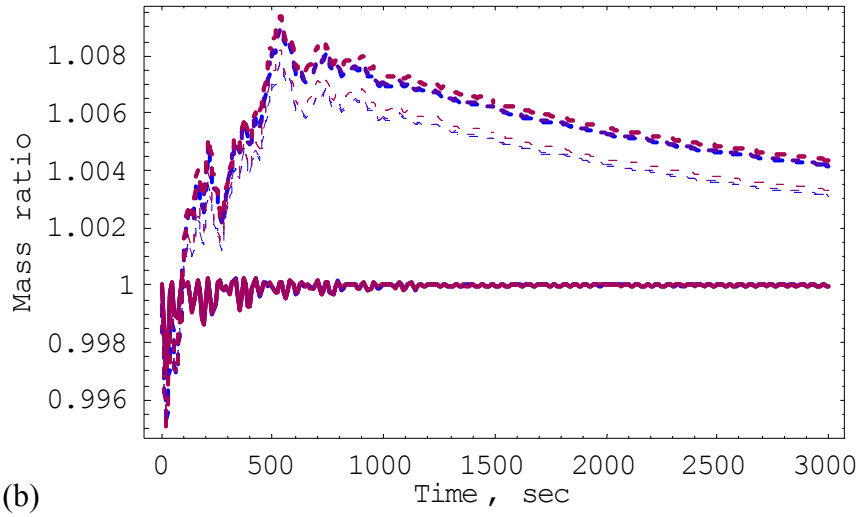
For the base diffusion / eddy viscosity combination, the higher gradient of 10.0 could not be simulated when 5 *skipsteps* were used, therefore the red curve is missing from Figure 5.64b. For the smaller gradient, no significant change is noted in either the DG or interpolated CG mass results, as *substeps* and *skipsteps* are varied. Meanwhile, for the larger gradient, a slight difference is visible in the various interpolated CG results, but not in the DG results. This parameter does not appear to have a significant impact on mass conservation.

Summary of parameter studies

For all of the parameter studies, the DG mass results are conservative for all parameter values, gradient strengths, and gradient type. Meanwhile, increasing the diffusion coefficients had the greatest impact and subsequent improvement in the interpolated CG mass results. For higher values of diffusion, acceptable mass ratios can be achieved for even the high gradients. Meanwhile, increasing the eddy viscosities does not significantly improve the mass balance of the model. The mass balance properties of the two interpolation methods did not vary significantly, however the interface average interpolation method has slightly better mass properties for the higher gradients. The low-



(a)



(b)

gradual gradients sharp gradients

DG results interpolated CG results

Figure 5.64 Mass ratios versus time for temporal parameter study: (a) gradient of 1.0 and (b) gradient of 10.0. Parameter combinations: base (*substeps* = 1 / *skipsteps* = 1), 5/1 (blue), 2/1 (purple), 1/2 (magenta), 1/5 (red).

order spatial approximations have slightly better mass properties than the higher-order approximations; and the temporal parameters, *substeps* and *skipsteps*, do not have any impact on mass conservation.

5.9 Conclusions and recommendations

5.9.1 1D results

In general, for both the x and z coordinates, near perfect mass balance can be achieved over the entire range of diffusion coefficients (advection-dominated to diffusion-dominated) when Type II or III boundary conditions are used. However, for a completely closed domain, Type III BCs are recommended since Type II BCs allow mass to escape when the plume reaches the boundary. Neither spatial nor temporal resolution appears to play a significant role in mass conservation, as any reasonable spatial or temporal resolution will provide good mass conservation. This verifies the local conservation property of the LDG algorithm. Additionally, for the verification test cases in the x and z coordinates, the algorithm converges to the analytical solution as the grid is refined. The spatial convergence rates for all of the test cases are summarized in Table 5.35.

Table 5.35 Spatial convergence rates for 1D verification test cases in the x- and z-coordinate directions.

		Convergence rates		
		peak	average	best-fit
Advection dominated	L_2	0.556 (1.034)	0.394 (0.555)	0.402 (0.541)
	L_∞	0.385 (0.347)	0.115 (0.205)	0.075 (0.205)
Diffusive advection dominated	L_2	2.302 (2.197)	1.893 (1.782)	1.863 (1.779)
	L_∞	1.823 (1.179)	1.563 (1.120)	1.571 (1.117)
Diffusive transition	L_2	1.982 (2.052)	1.837 (1.917)	1.853 (1.935)
	L_∞	2.052 (1.950)	1.658 (1.839)	1.615 (1.846)
Diffusion dominated	L_2	1.950 (1.929)	1.800 (1.794)	1.814 (1.813)
	L_∞	2.035 (2.146)	1.680 (1.551)	1.602 (1.574)

*First number is the x-coordinate rate and the number in parentheses is the z-coordinate rate.

These rates were computed from the linear portion of the convergence plots given in §5.7.1. Notice that for both the x- and z-coordinate directions, most of the convergence rates are near the theoretical value of 2 for linear approximating functions. The advection-dominated rates are reduced due to the sharpness of the front and the unavoidable fact that an instantaneous jump is being approximated with several data points. Furthermore, notice that for each row in the table, which corresponds to a certain type of simulation, i.e., advection dominated to diffusion dominated, the x- and z-coordinate convergence rates are similar; indicating that the transport algorithm itself converges uniformly in both spatial coordinates. Furthermore, no additional temporal accuracy is gained when the time step is reduced past $\Delta t = 0.1$ seconds.

Finally, for diffusion dominated problems, the stability of the algorithm was found to depend upon the Peclet number and was reduced significantly from the maximum stable Courant number of 0.5, such that the maximum stable Courant number was approximately equal to one tenth of the Peclet number.

5.9.2 2D results

From the constant velocity tests, it is apparent that the transport model is capable of handling sharp fronts and large gradients while retaining mass conservation. Furthermore, the mass results from the full baroclinic model, which were computed from the DG salinity fields, are also conservative in an "average" sense; where "average" is used to indicate that the mass ratio oscillates about 1.0 due to the changes in the sea surface elevation. However, the DG to CG interpolation is not conservative, and the mass computed from the interpolated CG salinity fields is not conserved. Therefore, the interpolation method should be examined further. Most importantly, it is obvious that the LDG salinity results from the coupled transport/hydrodynamic baroclinic model are conserving mass.

5.9.3 General recommendations

The LDG discretization of the advection-dispersion equation adds the required mass conservative transport capabilities to the existing ADCIRC hydrodynamic model in order to create a full baroclinic model. Additionally, the LDG transport model is capable of modeling both advective and diffusive flows, while retaining mass conservation, and is a promising addition to the ADCIRC model. The traditional continuous discretizations that were compared to the LDG discretization in another study (*Atkinson et al*. [2004]) are not capable of handling highly advective flows and conserving mass.

However, the interpolation from the DG gridspace to the CG gridspace is not conservative, and should be examined further for stability and conservation properties. Furthermore, the extra interpolation computations that are necessary for grid communication between the two models introduce additional mechanisms whereby errors in the CG gridspace can build up over time. Finally, the temporal interplay between the ADCIRC hydrodynamics and the LDG transport module should be examined with more realistic gradients and physical bathymetry profiles, which will require the reintroduction of the Γ terms from the sigma coordinate transformation. These terms introduce more communication between the gridspaces and temporal interplay at several time levels, which further complicates the stability analysis of the combined model. Proper application of these terms is an area of continuing work.

CHAPTER 6

Summary of Important Findings and Unresolved Issues

6.1 Numerical propagation analysis tools

In Chapter 2, numerical approximations to dispersion and Fourier analysis were developed and verified against their analytical counterparts. Although limited to constant bathymetry, these tools are able to predict the correct analytical behavior, and can be used to study algorithms that otherwise could not be studied analytically for various reasons. Such algorithms include higher-order discontinuous methods with slope limiting and coupled continuous and discontinuous models. Unfortunately, no single combination of wave initialization and wave tracking techniques was able to correctly match both the dispersion and Fourier behavior for all of the algorithms; a combination of these techniques must be used. These tools provide a method for evaluating slope limiters within the finite volume and discontinuous Galerkin algorithms; namely, to examine the wave propagation properties of the limiters rather than simply choosing the first one that works. They also provide a mechanism for evaluating multi-algorithmic models.

6.2 Comparison of SWE solution algorithms

In Chapter 3, the staggered finite difference, primitive finite element, generalized wave continuity (ADCIRC model), selective lumping finite element, finite volume, and

discontinuous Galerkin (DG) finite element discretizations for the 1D linear SWE were compared. It was found that the relatively new DG methods have similar error and propagation properties, when compared to the often used ADCIRC model. Namely, near second-order spatial convergence and very little numerical dissipation of physical waves when higher-order interpolates are used. Additionally, the DG class of algorithms have many attractive properties, such as shock capturing without oscillations, variable h and p refinement, and easily modified grid spaces without hanging nodes and mortar spaces; although, they also require more computational effort. Thus a study of various coupling mechanisms for discontinuous and continuous finite element algorithms was undertaken.

6.3 Coupled continuous and discontinuous models

In Chapter 4, the properties of subdomain and equation multi-algorithmic models for the SWE were examined. In subdomain multi-algorithmic models, different algorithms are employed in unique partitions, or subdomains, of the larger domain in order to exploit each algorithm's best features, while minimizing their weaknesses. For equation multi-algorithmic models, different algorithms are used to discretize the continuity and momentum equations within the same domain. Propagation analysis of the three coupling schemes indicate that all three are capable of propagating a wave with minimal distortion; however, the low-order versions overdamp the physical waves and the unlimited higher-order versions of the equation coupled schemes allow wiggles in the solutions. Furthermore, it was found that the equation coupling was often unstable and not well suited for actual applications. Additionally, results from this simplified study indicate that the subdomain interface must be located a suitable distance away from the shelf break, where the highly advective behavior resides. Meanwhile, the subdomain coupled $\mathrm{DG} \leftrightarrow \mathrm{ADCIRC}$ model shows promise, but flux coupling with a full non-linear model should be examined for the interface coupling rather than overlapping elements. Additional work could be done using mass balance to examine the optimal location of the subdomain

interface, as well as exploring the optimal load balance due to temporal constraints within each subdomain. Finally, a second-order Runge-Kutta temporal scheme should be used for the second-order spatial approximations (piecewise linears), such that the spatial and temporal discretizations are of the same order.

6.4 LDG transport module for baroclinic ADCIRC model

In Chapter 5, as a first step toward a full three dimensional baroclinic model, a quasi-3D (*x*-*z* slice) DG transport algorithm was developed and incorporated into a baroclinic ADCIRC hydrodynamic model using the same coordinates. The LDG transport algorithm was verified by comparing the model results to the analytical solution for 1D breakthrough curves. Additionally, the mass conservation properties of the model were examined by simulating Gauss plumes in each coordinate direction. From these verification tests, the algorithm was found to be stable and mass conservative over the range of advection-dominated to diffusion-dominated flow regimes. Furthermore, 2D salinity "dam break" tests of the full baroclinic ADCIRC/LDG transport model revealed that the model is stable and mass conservative (for the DG salinity field) for this difficult test case. However, the interpolation method that is used to translate the DG salinity field into the CG gridspace, which is required by the ADCIRC model, is not mass conservative. Therefore, more work needs to be done to improve the conservation properties of the interpolation step. Additionally, a second-order Runge-Kutta temporal scheme should be used when linear spatial interpolants are utilized. Finally, for applications with non-constant bathymetry, the Γ terms from the sigma coordinate transformation will have to be introduced. Preliminary tests show that these terms introduce stability issues and eliminate the mass conservation properties of the LDG transport module. Therefore, further work needs to be done in order to determine the proper implementation of these terms.

References

Aizinger V, Dawson C, Cockburn B, Castillo P, 2001. The local discontinuous Galerkin method for contaminant transport. *Advances in Water Resources*; 24: 73-87.

Aizinger V, Dawson C, 2002. A discontinuous Galerkin method for two-dimensional flow and transport in shallow water. *Advances in Water Resources*; 25: 67-84.

Alcrudo F, Garcia-Navarro P, 1993. A high-resolution Godunov-type scheme in finite volumes for the 2D shallow water equations. *Numerical Methods in Fluids*; 16: 489-505.

Atkinson JH, Westerink JJ, Luettich RA, 2004. Two-dimensional dispersion analysis of finite element approximations to the shallow water equations. *International Journal for Numerical Methods in Fluids*; 45(7): 715-749.

Atkinson JH, Szpilka CM, Kolar RL, 2004. A comparison of continuous and discontinuous Galerkin methods for hydrodynamic transport. *Proceedings of the XV International Conference on Computational Methods in Water Resources*; UNC-Chapel Hill. C Miller et al. (eds), CDROM.

Bassi F, Rebay S, 1997. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *Journal of Computational Physics*; 131: 267-279.

Bell JB, Shubin GR, 1984. *Higher order godunov methods for reservoir simulation*. Exxon Production Research Company Technical Report EPR.12PR.84, Houston.

Causon DM, Ingram DM, Mingham CG, Yang G, Pearson RV, 2000. Calculation of shallow water flows using a cartesian cut cell approach. *Advances in Water Resources*; 23(5): 545-562.

Celia MA, Gray WG, 1992. Accuracy and error reduction. In *Numerical Methods for Differential Equations*, Prentice-Hall, New Jersey; pp 330-359.

Chippada S, Dawson CN, Martinez ML, Wheeler MF, 1998. A Godunov-type finite volume method for the system of shallow water equations. *Computer Methods in Applied Mechanics & Engineering*; 151(1-2): 105-129.

Cockburn B, Hou S, Shu C-W, 1990. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case. *Mathematics of Computation*; 54(190): 545-581.

Cockburn B, Karniadakis GE, Shu, C-W, 2002. The development of discontinuous Galerkin methods. In *Discontinuous Galerkin Methods*, Cockburn et al. (eds.), vol 11, Springer-Verlag, Berlin; 3-50.

Cockburn B, Lin S-Y, Shu C-W, 1989. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems.

Journal of Computational Physics; 84: 90-113.

Cockburn B, Shu C-W, 1989. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework. *Mathematics of Computation*; 52(186): 411-435.

Cockburn B, Shu C-W, 1991. The Runge-Kutta local projection P^1-discontinuous-Galerkin finite element method for scalar conservation laws. *Mathematical Modeling and Numerical Analysis*; 25(3): 337-361.

Cockburn B, Shu C-W, 1998a. The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems. *Journal of Computational Physics*; 141: 199-224.

Cockburn B, Shu C-W, 1998b. The local discontinuous Galerkin finite element method for convection-diffusion systems. *SIAM Journal of Numerical Analysis*; 35(6): 2440-2463.

Cushman-Roisin B, 1994. The governing equations. In *Introduction to Geophysical Fluid Dynamics*, Prentice Hall, New Jersey; pp 33-46.

Dawson C, Proft J, 2001. A priori error estimates for interior penalty versions of the local discontinuous Galerkin method applied to transport equations. *Numerical Methods for Partial Differential Equations*; 17(6): 545-564.

Dawson C, Proft J, 2002a. Coupling of continuous and discontinuous Galerkin methods for transport problems. *Computer Methods in Applied Mechanics and Engineering*; 191: 3213-3231.

Dawson C, Proft J 2002b. Discontinuous and coupled continuous/discontinuous Galerkin methods for the shallow water equations. *Computer Methods in Applied Mechanics and Engineering*; 191: 4721-4746.

Dawson C, Proft J. 2004. Coupled discontinuous and continuous Galerkin finite element methods for shallow water equations. *Computer Methods in Applied Mechanics and Engineering*; 193: 289-318.

Dresback KM, Kolar RL, 2004a. Truncation error analysis of ADCIRC. EM/GIS Technical Report #04-02. Available from: http://www.coe.ou.edu/emgis/kolar/pages/reports.html.

Dresback KM, Kolar RL, 2004b. Reference manual for the 2D *x-z* ADCIRC code. EM/GIS Technical Report #04-03. Available from: http://www.coe.ou.edu/emgis/kolar/pages/reports.html.

Dresback KM, Kolar RL, Dietrich JC, 2005. On the form of the momentum equation for shallow water models based on the generalized wave continuity equation. *Advances in Water Resources*; 28: 345-358.

Foreman MGG, 1983. An analysis of the "wave equation" model for finite element tidal computations. *Journal of Computational Physics*; 52: 290-312.

Foreman MGG, 1984. A two-dimensional dispersion analysis of selected methods for solving the linearized shallow water equations. *Journal of Computational Physics*; 56:287-323.

Gray WG, Lynch DR, 1977. Time-stepping schemes for finite element tidal model computations. *Advances in Water Resources*; 1(2): 83-95.

Haidvogel D, 1999. *Numerical ocean circulation modeling.* Beckman Imperial College Press, London; 320pp.

Hench JL, Luettich RA, 1997. Analysis and application of Eulerian finite element methods for the transport equation. In ECM-5.

Hood P, Taylor C, 1974. Navier-Stokes equations using mixed interpolation. In *Finite Element Methods in Flow Problems*, Oden et al. (eds.), U. Alabama Press, Huntsville, AL; 121-132.

Hu FQ, Hussaini MY, Rasetarinera P, 1999. An analysis of the discontinuous Galerkin method for wave propagation problems. *Journal of Computational Physics*; 151: 921-946.

Kawahara M, Hirano H, Tsubota K, Inagaki K 1982. Selective lumping finite element method for shallow water flow. *International Journal for Numerical Methods in Fluids*; 2: 89-112.

Kinnmark I, 1986. *The shallow water wave equations: formulation, analysis, and application.* Lecture notes in engineering, vol 15, Springer-Verlag, Berlin.

Kinnmark I, Gray WG, 1985. The 2Δx-test: a tool for analyzing spurious oscillations. *Advances in Water Resources*; 8: 129-135.

Kolar RL, 1992. *Environmental Conservation Laws: Formulation, Numerical Solution, and Application*. Ph.D. Dissertation, Dept. of Civil Engineering, University of Notre Dame, IN; 297 pp.

Kolar RL, Westerink JJ, Cantekin ME, Blain CA, 1994. Aspects of nonlinear simulations using shallow-water models based on the wave continuity equation. *Computers and Fluids*; 23(3): 523-538.

Knox RC, Sabatini DA, Canter LW, 1993. Modeling subsurface transport and fate processes. In *Subsurface Transport and Fate Processes*, Lewis Publishers, Florida; pp 283-352.

Leendertse JJ, 1967. *Aspects of a computational model for long period water-wave propagation.* Rand Corporation Technical Report RM-5294-PR, Santa Monica.

LeVeque, RJ, 1990. *Numerical Methods for Conservation Laws*. Birkhauser Verlag, Basel.

Luettich RA, 2002. *Personal communication*.

Luettich RA, Westerink JJ, Scheffner NW, 1992. ADCIRC: an Advanced Three-Dimensional Circulation Model for Shelves, Coasts and Estuaries; Report 1: Theory and Methodology of ADCIRC-2DDI and ADCIRC-3DL. Technical Report DRP-92-6, Dept. of the Army, USACE, Washington, DC.

Luettich RA, Westerink JJ, 2003. Formulation and Numerical Implementation of the 2D/3D ADCIRC Finite Element Model Version 43.XX, Technical Report. Available on the web at www.unc.edu/depts/marine/C_CATS/adcirc/.

Lynch DR, Gray WG, 1979. A wave equation model for finite element tidal computations. *Computers and Fluids*; 7(3): 207-228.

Oden JT, Babuska I, Baumann CE, 1998. A discontinuous *hp* finite element method for diffusion problems. *Journal of Computational Physics*; 146: 491-519.

Platzman GW, 1981. Some response characteristics of finite-element tidal models. *Journal of Computational Physics*; 40: 36-63.

Proft JK, 2002. *Multi-Algorithmic Numerical Strategies for the Solution of Shallow Water Models*. Ph.D. Dissertation, University of Texas at Austin, Austin, TX; 157 pp.

Raviart RA, Thomas JM, 1986. A mixed finite element method for 2nd order elliptic problems. In *Mathematical Aspects of the Finite Element Method*, Lecture notes in mathematics, vol 606, Springer-Verlag; 292-315.

Rivière B, Wheeler M, Girault V, 1999. Improved energy estimates for interior penalty, constrained and discontinuous Galerkin methods for elliptic problems part I. *Computational Geosciences*; 3: 337-360.

Rivière B, Wheeler MF, Girault V, 2001. A priori error estimates for finite element methods based on discontinuous approximation spaces for elliptic problems. *SIAM Journal of Numerical Analysis*; 39(3): 902-931.

Roache P, 1998. *Fundamentals of computational fluid dynamics*. Hermosa Publishers, New Mexico; 648 pp.

Roe PL, 1981. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*; 43: 357-372.

Siegel P, Mosé R, Ackerer PH, Jaffre J, 1997. Solution of the advection-diffusion equation using a combination of discontinuous and mixed finite elements. *Journal for Numerical Methods in Fluids*; 24: 595-613.

Sweby PK, 1984. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM Journal of Numerical Analysis*; 21(5): 995-1011.

Vichnevetsky R, Shieh YS, 1972. *On the numerical method of lines for one dimensional water quality equations*. Dept. of Computer Science Technical Report #20, Rutgers University.

Walters RA, 1983, Numerically induced oscillations in finite element approximations to the shallow water equations. *International Journal for Numerical Methods in Fluids*; 3: 591-604.

Westerink JJ, Luettich RA, Wu JK, Kolar RL, 1994. The influence of normal flow boundary conditions on spurious modes in finite element solutions to the shallow water equations. *International Journal for Numerical Methods in Fluids*; 18: 1021-1060.

APPENDIX A

Derivation of Continuous Finite Element Discretizations of the Shallow Water Equations

A.1 Linearized shallow water equations in 1D

The system of linear SWE in 1D consists of the primitive continuity equation

$$\zeta_t + hu_x = 0 \qquad \textbf{(A.1)}$$

and the inviscid momentum equation

$$u_t + \tau u + g\zeta_x = 0\,, \qquad \textbf{(A.2)}$$

where u is the depth-averaged velocity, ζ is the surface elevation, τ is the linear bottom friction factor, g is the acceleration of gravity, h is the bathymetric depth of water (assumed constant), and the subscripts indicate partial derivatives. The linearized form of the generalized wave continuity (GWC) equation is given as

$$\zeta_{tt} + G\zeta_t + (G-\tau)hu_x - gh\zeta_{xx} = 0 \qquad \textbf{(A.3)}$$

where G is a numerical coefficient that determines the balance between primitive (large values of G) and pure wave (small values of G) forms. The above equations are also presented in Chapter 1 (§1.2.2). All of the algorithms presented in the following sections use the same grid scheme, which is shown below in Figure A.1.

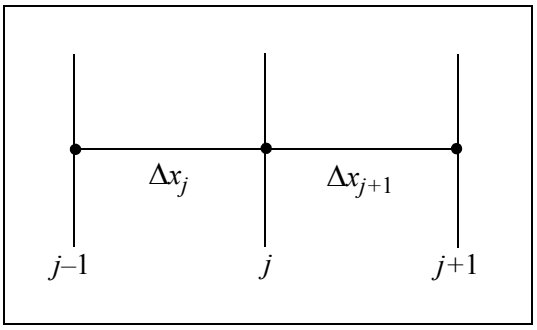


Figure A.1 Discrete grid representation for continuous finite element algorithms.

A.2 Primitive finite element (leap-frog)

The starting point of a traditional finite element discretization to the SWE is to multiply Equations (A.1) and (A.2) by a weight function, ϕ_i, and integrate over the domain, Ω. This gives the weighted residual form as

$$(\zeta_t,\phi_i)_\Omega + (hu_x,\phi_i)_\Omega = 0 \qquad \textbf{(A.4)}$$

$$(u_t,\phi_i)_\Omega + (\tau u,\phi_i)_\Omega + (g\zeta_x,\phi_i)_\Omega = 0 \qquad \textbf{(A.5)}$$

where the inner product notation, $(\ ,\ )_\Omega$, denotes integration over the domain.

Next, the variables are approximated by some known function; in a Galerkin finite element discretization, the approximating functions are the same as the weight functions. For linear Lagrange functions, the approximations are given as

$$\zeta^h = \sum_{j=1}^{M} \zeta_j(t)\phi_j(x) \qquad \textbf{(A.6)}$$

$$u^h = \sum_{j=1}^{M} u_j(t)\phi_j(x) \qquad \textbf{(A.7)}$$

where M is the number of nodes in the domain and the $\phi_j(x)$ are standard linear chapeau functions defined for node j as

$$\phi_j(x) = (x - x_{j-1})/(x_j - x_{j-1}) \qquad x_{j-1} \le x \le x_j \tag{A.8}$$

$$\phi_j(x) = (x_{j+1} - x)/(x_{j+1} - x_j) \qquad x_j \le x \le x_{j+1} \ . \tag{A.9}$$

Substitution of these approximations gives a system of $2M$ equations for $2M$ unknowns:

$$\sum_{j=1}^{M} \left[\frac{\partial}{\partial t}(\zeta_j \phi_j, \phi_i)_\Omega + (h u_j \frac{\partial \phi_j}{\partial x}, \phi_i)_\Omega = 0 \right] \tag{A.10}$$

$$\sum_{j=1}^{M} \left[\frac{\partial}{\partial t}(u_j \phi_j, \phi_i)_\Omega + (\tau u_j \phi_j, \phi_i)_\Omega + (g \zeta_j \frac{\partial \phi_j}{\partial x}, \phi_i)_\Omega = 0 \right] . \tag{A.11}$$

On a master element, this gives

$$\mathbf{M}_{i,j} \frac{\partial \zeta_j}{\partial t} + h \mathbf{B}_{i,j} u_j = 0 \tag{A.12}$$

$$\mathbf{M}_{i,j} \frac{\partial u_j}{\partial t} + \tau \mathbf{M}_{i,j} u_j + g \mathbf{B}_{i,j} \zeta_j = 0 \tag{A.13}$$

where

$$\mathbf{M}_{i,j} = (\phi_j, \phi_i)_{\Omega_e} = \Delta x_j \begin{bmatrix} 1/3 & 1/6 \\ 1/3 & 1/6 \end{bmatrix}$$

$$\mathbf{B}_{i,j} = \left(\frac{\partial \phi_j}{\partial x}, \phi_i \right)_{\Omega_e} = \begin{bmatrix} -1/2 & 1/2 \\ -1/2 & 1/2 \end{bmatrix} .$$

For a uniform grid, this results in the following representations at an interior node *j*:

$$\frac{1}{6}\frac{d}{dt}[\zeta_{j-1}+4\zeta_j+\zeta_{j+1}]+\frac{h}{2\Delta x}[u_{j+1}-u_{j-1}] = 0 \qquad \textbf{(A.14)}$$

$$\frac{1}{6}\frac{d}{dt}[u_{j-1}+4u_j+u_{j+1}]+\frac{g}{2\Delta x}[\zeta_{j+1}-\zeta_{j-1}]+\frac{\tau}{6}[u_{j-1}+4u_j+u_{j+1}] = 0\ . \qquad \textbf{(A.15)}$$

In a leap-frog temporal discretization, the temporal derivatives are approximated by centered differences at time levels k+1 and k–1. Meanwhile the spatial derivatives are evaluated at level k and the bottom friction is evaluated at level k–1 to improve stability (*Gray and Lynch*, [1977]). After multiplying through by 2Δt, this gives the final discretization as

$$\zeta_{j-1}^{k+1}+4\zeta_j^{k+1}+\zeta_{j+1}^{k+1} = (\zeta_{j-1}^{k-1}+4\zeta_j^{k-1}+\zeta_{j+1}^{k-1})-6h\frac{\Delta t}{\Delta x}(u_{j+1}^{k}-u_{j-1}^{k}) \qquad \textbf{(A.16)}$$

$$u_{j-1}^{k+1}+4u_j^{k+1}+u_{j+1}^{k+1} = (1-2\tau\Delta t)(u_{j-1}^{k-1}+4u_j^{k-1}+u_{j+1}^{k-1})-6g\frac{\Delta t}{\Delta x}(\zeta_{j+1}^{k}-\zeta_{j-1}^{k})\,, \qquad \textbf{(A.17)}$$

where it is evident that this temporal discretization serves to decouple the continuity and momentum equations. The general equations for an interior node that are presented above are needed for the analysis in Chapter 3; however, in practice global matrices are built up from the master elements and the entire system is solved at once. This would result in two tridiagonal systems; one for the elevations and one for the velocities.

A.3 Generalized wave continuity (ADCIRC model)

For the GWC (or ADCIRC) model, the generalized wave continuity equation (A.3) is solved with the momentum equation (A.2) instead of solving the primitive continuity equation (A.1). As in the primitive finite element model, this first step is to multiply by a weight function ϕ_i, and integrate over the domain, Ω to get

$$(\zeta_{tt},\phi_i)_\Omega + (G\zeta_t,\phi_i)_\Omega + ((G-\tau)hu_x,\phi_i)_\Omega + (gh\zeta_x,\frac{\partial\phi_i}{\partial x})_\Omega - gh\zeta_x\Big|_{x_0}^{x_L} = 0 \quad \textbf{(A.18)}$$

$$(u_t,\phi_i)_\Omega + (\tau u,\phi_i)_\Omega + (g\zeta_x,\phi_i)_\Omega = 0 \quad \textbf{(A.19)}$$

where the GWC equation has been integrated by parts and $[x_0, x_L]$ are the limits of the domain.

Next substitute the approximations for ζ and u, given by Equations (A.6) and (A.7), to get

$$\sum_{j=1}^{M}\left[\frac{\partial^2}{\partial t^2}(\zeta_j\phi_j,\phi_i)_\Omega + \frac{\partial}{\partial t}(G\zeta_j\phi_j,\phi_i)_\Omega + ((G-\tau)hu_j\frac{\partial\phi_j}{\partial x},\phi_i)_\Omega + (gh\zeta_j\frac{\partial\phi_j}{\partial x},\frac{\partial\phi_i}{\partial x})_\Omega\right] = B \quad \textbf{(A.20)}$$

$$\sum_{j=1}^{M}\left[\frac{\partial}{\partial t}(u_j\phi_j,\phi_i)_\Omega + (\tau u_j\phi_j,\phi_i)_\Omega + (g\zeta_j\frac{\partial\phi_j}{\partial x},\phi_i)_\Omega = 0\right] \quad \textbf{(A.21)}$$

where the boundary term, B, is equal to $gh\zeta_x\Big|_{x_0}^{x_L}$.

On a master element this gives

$$\mathbf{M}_{i,j}\frac{\partial^2\zeta_j}{\partial t^2} + G\mathbf{M}_{i,j}\frac{\partial\zeta_j}{\partial t} + (G-\tau)h\mathbf{B}_{i,j}u_j + gh\mathbf{S}_{i,j}\zeta_j = B \quad \textbf{(A.22)}$$

$$\mathbf{M}^L_{i,j}\frac{\partial u_j}{\partial t} + \tau\mathbf{M}^L_{i,j}u_j + g\mathbf{B}_{i,j}\zeta_j = 0 \quad \textbf{(A.23)}$$

where the mass matrix for the momentum equation is lumped to decouple the velocity unknowns and the boundary terms, B, only enter in on the boundary elements. Matrices $\mathbf{M}_{i,j}$ and $\mathbf{B}_{i,j}$ are as given above for the primitive scheme, and the remaining matrices are given as

$$\mathbf{M}^{L}_{i,j} = (\phi_j, \phi_i)^{L}_{\Omega_e} = \Delta x_j \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix}$$

$$\mathbf{S}_{i,j} = (\frac{\partial \phi_j}{\partial x}, \frac{\partial \phi_i}{\partial x})_{\Omega_e} = \frac{1}{\Delta x_j} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} .$$

Following the ADCIRC model (*Luettich et al.* [1992, 2003]), the temporal discretization utilizes a two-level scheme centered at $k+1/2$ for the momentum and a three-level scheme centered at k for the GWC as follows:

$$\frac{1}{\Delta t^2}\mathbf{M}_{i,j}(\zeta_j^{k+1} - 2\zeta_j^{k} + \zeta_j^{k-1}) + \frac{G}{2\Delta t}\mathbf{M}_{i,j}(\zeta_j^{k+1} - \zeta_j^{k-1}) + (G-\tau)\mathbf{B}_{i,j} h u_j^{k} +$$

$$gh\mathbf{S}_{i,j}(\alpha_1\zeta_j^{k+1} + \alpha_2\zeta_j^{k} + \alpha_3\zeta_j^{k-1}) = B^{k} \qquad \textbf{(A.24)}$$

$$\frac{1}{\Delta t}\mathbf{M}^{L}_{i,j}(u_j^{k+1} - u_j^{k}) + \frac{\tau}{2}\mathbf{M}^{L}_{i,j}(u_j^{k+1} + u_j^{k}) + \frac{g}{2}\mathbf{B}_{i,j}(\zeta_j^{k+1} + \zeta_j^{k}) = 0 \qquad \textbf{(A.25)}$$

where the time weighting parameters, α_n, in the GWC equation must satisfy $\alpha_1 + \alpha_2 + \alpha_3 = 1.0$, and typically all three values are set equal to 1/3. Equation (A.24) is solved for the new elevation values and then these are used in Equation (A.25) to find the new velocity values. For an interior node, j, the discrete equations are

$$\left(\frac{1}{\Delta t^2} + \frac{G}{2\Delta t}\right)\frac{\Delta x}{6}(\zeta_{j-1}^{k+1} + 4\zeta_j^{k+1} + \zeta_{j+1}^{k+1}) + \frac{\alpha_1 gh}{\Delta x}(-\zeta_{j-1}^{k+1} + 2\zeta_j^{k+1} - \zeta_{j+1}^{k+1}) =$$

$$\frac{2\Delta x}{6\Delta t^2}(\zeta_{j-1}^{k} + 4\zeta_j^{k} + \zeta_{j+1}^{k}) + \left(-\frac{1}{\Delta t^2} + \frac{G}{2\Delta t}\right)\frac{\Delta x}{6}(\zeta_{j-1}^{k-1} + 4\zeta_j^{k-1} + \zeta_{j+1}^{k-1}) -$$

$$(G-\tau)\frac{h}{2}(u_{j+1}^{k} - u_{j-1}^{k}) - \frac{\alpha_2 gh}{\Delta x}(-\zeta_{j-1}^{k} + 2\zeta_j^{k} - \zeta_{j+1}^{k}) -$$

$$\frac{\alpha_3 gh}{\Delta x}(-\zeta_{j-1}^{k-1} + 2\zeta_j^{k-1} - \zeta_{j+1}^{k-1}) \qquad \textbf{(A.26)}$$

$$\Delta x\left(1+\frac{\tau\Delta t}{2}\right)u_j^{k+1} = \Delta x\left(1-\frac{\tau\Delta t}{2}\right)u_j^{k} - \frac{g\Delta t}{4}(\zeta_{j+1}^{k+1}+\zeta_{j+1}^{k}-(\zeta_{j-1}^{k+1}+\zeta_{j-1}^{k})). \quad \textbf{(A.27)}$$

The general interior node equations presented above are needed for the analysis in Chapter 3. However, in practice a global matrix is built up using the master elements and the entire system is solved at once. Notice that the momentum discretization is uncoupled due to the lumping and that the GWC discretization results in a tridiagonal system, which can easily be solved without using a full matrix solver.

A.4 Selective lumping finite element

The SLFE scheme follows the primitive finite element scheme for the spatial discretization – through Equations (A.12) and (A.13) for a master element, which are repeated here for convenience

$$\mathbf{M}_{i,j}\frac{\partial\zeta_j}{\partial t} + h\mathbf{B}_{i,j}u_j = 0 \quad \textbf{(A.28)}$$

$$\mathbf{M}_{i,j}\frac{\partial u_j}{\partial t} + \tau\mathbf{M}_{i,j}u_j + g\mathbf{B}_{i,j}\zeta_j = 0 \ . \quad \textbf{(A.29)}$$

At the point of the temporal discretization, the SLFE scheme departs from the PFE. Rather than a single step with leap-frog time stepping to decouple the elevation and velocity solutions, the SLFE uses an explicit two-step scheme wherein the mass matrices are selectively lumped.

The scheme can take one of two forms, depending upon whether the "viscous" terms are included. The inclusion of these terms at the *k* time level serves to stabilize the algorithm, and when they are excluded, the mass matrices at the *k* time level in the momentum equation must be selectively lumped, as is done in the continuity equation, in order to maintain stability. In this study, the selective lumping has been further generalized

by adding a separate lumping parameter for the bottom friction term, since the second-order viscous terms are lost in the linearization.

In this form, all of the mass matrices at time level *k* are selectively lumped, with the bottom friction remaining separate, such that the selective lumping scheme can be expressed as follows for the first half step:

$$\mathbf{M}^{L}_{i,j}\zeta_j^{k+1/2} = \mathbf{M}^{e}_{i,j}\zeta_j^{k} - \frac{\Delta t}{2}h\mathbf{B}_{i,j}u_j^{k} \tag{A.30}$$

$$\mathbf{M}^{L}_{i,j}u_j^{k+1/2} = \mathbf{M}^{e}_{i,j}u_j^{k} - \frac{\Delta t}{2}(g\mathbf{B}_{i,j}\zeta_j^{k} + \tau\mathbf{M}^{f}_{i,j}u_j^{k}) \tag{A.31}$$

and for the second half step

$$\mathbf{M}^{L}_{i,j}\zeta_j^{k+1} = \mathbf{M}^{e}_{i,j}\zeta_j^{k} - \Delta t h\mathbf{B}_{i,j}u_j^{k+1/2} \tag{A.32}$$

$$\mathbf{M}^{L}_{i,j}u_j^{k+1} = \mathbf{M}^{e}_{i,j}u_j^{k} - \Delta t(g\mathbf{B}_{i,j}\zeta_j^{k+1/2} + \tau\mathbf{M}^{f}_{i,j}u_j^{k+1/2}) \tag{A.33}$$

where $M^{L}_{i,j}$ is fully lumped and $M^{e}_{i,j}$ and $M^{f}_{i,j}$ are selectively lumped. The selectively lumped matrices are derived from a linear combination of the lumped and unlumped matrices and are given as

$$M^{e}_{i,j} = eM^{L}_{i,j} + (1-e)M_{i,j} = \frac{\Delta x_j}{6}\begin{bmatrix} 2+e & 1-e \\ 1-e & 2+e \end{bmatrix} \tag{A.34}$$

where *e* is the selective lumping parameter and expresses the ratio of the lumping from fully lumped ($e = 1$) to unlumped ($e = 0$). For the bottom friction term, $M^{f}_{i,j}$, simply replace *e* with *f* to get the selectively lumped matrix. The selective lumping technique was found by *Kawahara et al.* [1982] to stabilize the algorithm in time and reduce the numerical damping effect.

For an interior node, *j*, the discrete equations for the first half step are given as

$$\zeta_j^{k+1/2} = \frac{1-e}{6}\zeta_{j-1}^k + \frac{2+e}{3}\zeta_j^k + \frac{1-e}{6}\zeta_{j+1}^k - \frac{\Delta t}{4\Delta x}h(u_{j+1}^k - u_{j-1}^k) \quad \textbf{(A.35)}$$

$$u_j^{k+1/2} = \frac{1-e}{6}u_{j-1}^k + \frac{2+e}{3}u_j^k + \frac{1-e}{6}u_{j+1}^k - \frac{\Delta t}{4\Delta x}g(\zeta_{j+1}^k - \zeta_{j-1}^k) -$$

$$\frac{\tau\Delta t}{2}\left(\frac{1-f}{6}u_{j-1}^k + \frac{2+f}{3}u_j^k + \frac{1-f}{6}u_{j+1}^k\right) \quad \textbf{(A.36)}$$

and for the second half step

$$\zeta_j^{k+1} = \frac{1-e}{6}\zeta_{j-1}^k + \frac{2+e}{3}\zeta_j^k + \frac{1-e}{6}\zeta_{j+1}^k - \frac{\Delta t}{2\Delta x}h(u_{j+1}^{k+1/2} - u_{j-1}^{k+1/2}) \quad \textbf{(A.37)}$$

$$u_j^{k+1} = \frac{1-e}{6}u_{j-1}^k + \frac{2+e}{3}u_j^k + \frac{1-e}{6}u_{j+1}^k - \frac{\Delta t}{2\Delta x}g(\zeta_{j+1}^{k+1/2} - \zeta_{j-1}^{k+1/2}) -$$

$$\tau\Delta t\left(\frac{1-f}{6}u_{j-1}^{k+1/2} + \frac{2+f}{3}u_j^{k+1/2} + \frac{1-f}{6}u_{j+1}^{k+1/2}\right). \quad \textbf{(A.38)}$$

If the first step is substituted into the second step and like terms are gathered, a closed form for the interior nodes can be written as

$$\zeta_j^{k+1} = \left[\frac{2+e}{3} - \frac{\mu^2 gh}{4}\right]\zeta_j^k + \frac{1-e}{6}(\zeta_{j-1}^k + \zeta_{j+1}^k) + \frac{\mu^2 gh}{8}(\zeta_{j-2}^k + \zeta_{j+2}^k) -$$

$$\frac{\mu h}{2}\left[\frac{1-e}{6} - \frac{\tau\Delta t}{2}\frac{1-f}{6}\right](u_{j+2}^k - u_{j-2}^k) - \frac{\mu h}{2}\left[\frac{2+e}{3} - \frac{\tau\Delta t}{2}\frac{2+f}{3}\right](u_{j+1}^k - u_{j-1}^k) \quad \textbf{(A.39)}$$

$$u_j^{k+1} = \left[\frac{\mu^2 gh}{8} - \tau\Delta t\frac{1-e}{6}\frac{1-f}{6} + \frac{\tau^2\Delta t^2}{2}\frac{(1-f)^2}{36}\right](u_{j-2}^k + u_{j+2}^k) +$$

$$\left[\frac{1-e}{6} - \tau\Delta t\left(\frac{1-f}{6}\frac{2+e}{3} + \frac{2+f}{3}\frac{1-e}{6} - \tau\Delta t\frac{1-f}{6}\frac{2+f}{3}\right)\right](u_{j-1}^k + u_{j+1}^k) +$$

$$\left[\frac{2+e}{3} - \frac{\mu^2 gh}{4} + 2\tau\Delta t\frac{1-f}{6}\left(\frac{\tau\Delta t}{2}\frac{1-f}{6} - \frac{1-e}{6}\right) + \tau\Delta t\frac{2+f}{3}\left(\frac{\tau\Delta t}{2}\frac{2+f}{3} - \frac{2+e}{3}\right)\right]u_j^k -$$

$$\frac{\mu g}{2}\left[\frac{1-e}{6}-\frac{\tau\Delta t}{2}\frac{1-f}{6}\right](\zeta_{j+2}^{k}-\zeta_{j-2}^{k})-\frac{\mu g}{2}\left[\frac{2+e}{3}-\frac{\tau\Delta t}{2}\frac{2+f}{3}\right](\zeta_{j+1}^{k}-\zeta_{j-1}^{k}) \qquad \textbf{(A.40)}$$

where $\mu = \Delta t/\Delta x$ and Equations (A.39) and (A.40) are not solved in practice, but are only given as a closed form herein for the analytical analyses in Chapter 3.

Note that when $e = f = 1$ and the solution is fully lumped, the general equations for an interior node simplify to

$$\zeta_{j}^{k+1} = \left[1-\frac{\mu^{2}gh}{4}\right]\zeta_{j}^{k}+\frac{\mu^{2}gh}{8}(\zeta_{j-2}^{k}+\zeta_{j+2}^{k})-\frac{\mu h}{2}\left[1-\frac{\tau\Delta t}{2}\right](u_{j+1}^{k}-u_{j-1}^{k}) \qquad \textbf{(A.41)}$$

$$u_{j}^{k+1} = \frac{\mu^{2}gh}{8}(u_{j-2}^{k}+u_{j+2}^{k})+\left[1-\frac{\mu^{2}gh}{4}+\tau\Delta t\left(\frac{\tau\Delta t}{2}-1\right)\right]u_{j}^{k}-$$

$$\frac{\mu g}{2}\left[1-\frac{\tau\Delta t}{2}\right](\zeta_{j+1}^{k}-\zeta_{j-1}^{k}) \; . \qquad \textbf{(A.42)}$$

APPENDIX B

Derivation of Staggered Finite Difference and Discontinuous Discretizations of the Shallow Water Equations

B.1 Linearized shallow water equations in 1D

The system of linear SWE in 1D consists of the primitive continuity equation

$$\zeta_t + hu_x = 0 \qquad \textbf{(B.1)}$$

and the inviscid momentum equation

$$u_t + \tau u + g\zeta_x = 0\,, \qquad \textbf{(B.2)}$$

where u is the depth-averaged velocity, ζ is the surface elevation, τ is the linear bottom friction factor, g is the acceleration of gravity, h is the bathymetric depth of water (assumed constant), and the subscripts indicate partial derivatives.

B.2 Staggered finite difference

Leendertse's [1967] staggered finite difference (SFD) algorithm for the SWE differs from traditional finite difference schemes in that the velocity and elevation grids are staggered relative to one another. This essentially results in the velocities being evaluated at the "edges" while the elevations are evaluated at the "center" of a grid block. Figure B.1

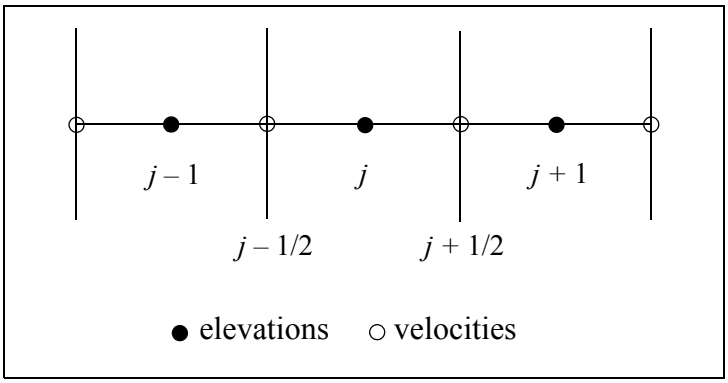


Figure B.1 Grid scheme for the SFD algorithm applied to the SWE.

illustrates the grid scheme for the SFD algorithm.

To derive the SFD discretization for the SWE, centered differences on the staggered grid are used to approximate the spatial derivatives. For Equation (B.1) this gives

$$\frac{\partial \zeta_j}{\partial t} + h\frac{(u_{j+1/2} - u_{j-1/2})}{\Delta x} = 0 \qquad \textbf{(B.3)}$$

and for Equation (B.2) this gives

$$\frac{\partial}{\partial t}u_{j+1/2} + \tau u_{j+1/2} + g\frac{(\zeta_{j+1} - \zeta_j)}{\Delta x} = 0\,, \qquad \textbf{(B.4)}$$

where the grid spacing is constant. The temporal discretization utilizes a weighted Euler scheme, giving the final approximations as

$$\zeta_j^{k+1} + h\theta\frac{\Delta t}{\Delta x}(u_{j+1/2}^{k+1} - u_{j-1/2}^{k+1}) = \zeta_j^k - h(1-\theta)\frac{\Delta t}{\Delta x}(u_{j+1/2}^k - u_{j-1/2}^k) \qquad \textbf{(B.5)}$$

$$(1 + \tau\Delta t\theta)u_{j+1/2}^{k+1} + g\theta\frac{\Delta t}{\Delta x}(\zeta_{j+1}^{k+1} - \zeta_j^{k+1}) =$$

$$(1-\tau\Delta t(1-\theta))u_{j+1/2}^{k}-g(1-\theta)\frac{\Delta t}{\Delta x}(\zeta_{j+1}^{k}-\zeta_{j}^{k}) \qquad \textbf{(B.6)}$$

where θ is the weighting parameter and varies between 0 (fully explicit) and 1 (fully implicit). A θ value of 0.5 gives the second-order Crank-Nicolson scheme.

B.3 Finite volume method

The FVM can utilize any order of interpolate within each discrete cell, allowing for local *p*-refinement, however piecewise constants (low-order) and piecewise linear interpolates are most common. Figure B.2 shows a schematic of a typical FVM discretization with piecewise linear interpolates, wherein each cell contains information about the cell-average value and the functional form of the slope for each variable. The arrows indicate the outward normal flux across the interfaces for cell *j*; notice that the interpolates need not be continuous across the interfaces.

The starting point of the FVM discretization is to rewrite Equations (B.1) and (B.2) as a system given as

$$\frac{\partial \mathbf{c}}{\partial t}+\frac{\partial \mathbf{f}}{\partial x}=\mathbf{h} \qquad \textbf{(B.7)}$$

where $\mathbf{c}=\begin{pmatrix}\zeta\\ u\end{pmatrix}$ is the vector of unknowns, $\mathbf{f}=\begin{pmatrix}hu\\ g\zeta\end{pmatrix}$ is the flux, and $\mathbf{h}=\begin{pmatrix}0\\ -\tau u\end{pmatrix}$. Next,

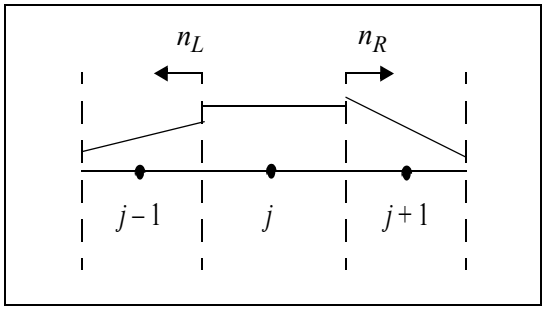


Figure B.2 Typical FVM discretization for control volume *j*.

after applying the divergence theorem, the weak integral form of the system is given as

$$\frac{\partial}{\partial t}\int_{\Omega_e} \mathbf{c}\, d\Omega_e + \int_{\Gamma_e} \mathbf{f}_n \cdot n_e d\Gamma_e = \int_{\Omega_e} \mathbf{h}\, d\Omega_e \qquad \textbf{(B.8)}$$

where Ω_e is the element area or volume, Γ_e is the element boundary, n_e is the unit outward normal and $\mathbf{f}_n$ is the normal flux across the boundary. In one spatial dimension, the boundary integrations are nothing more than point evaluations, such that for each volume the approximation is

$$\frac{\Delta x_j}{\Delta t}(\mathbf{c}_j^{k+1} - \mathbf{c}_j^k) + \mathbf{f}_L^k \cdot n_L + \mathbf{f}_R^k \cdot n_R = \Delta x_j \mathbf{h}_j^k \qquad \textbf{(B.9)}$$

where an explicit temporal discretization has been used.

Since the variables are allowed to be discontinuous across the element interfaces, the fluxes, $\mathbf{f}_L$ and $\mathbf{f}_R$, are not well defined. At each element interface, the Riemann problem must be solved. Note that it is generally not worthwhile to get an exact solution to the Riemann problem since roundoff and truncation will introduce more error into the solution than an approximate Riemann solver. Herein Roe's approximation is used to solve the Riemann problem for Equations (B.1) and (B.2), where the normal Jacobian matrix is defined by using the chain rule to rewrite the flux term in Equation (B.7) in an equivalent form

$$\frac{\partial \mathbf{f}}{\partial x} = \frac{\partial \mathbf{f}}{\partial \mathbf{c}}\frac{\partial \mathbf{c}}{\partial x} = \mathbf{A}(\mathbf{c}_L, \mathbf{c}_R)\frac{\partial \mathbf{c}}{\partial x} \qquad \textbf{(B.10)}$$

where, in general, **A** will be a function of the left and right states of **c**. However, for the linearized system of equations, **A** is independent of **c** as is shown here

$$\mathbf{A} = \frac{\partial \mathbf{f}}{\partial \mathbf{c}} = \begin{bmatrix} 0 & h \\ g & 0 \end{bmatrix} . \qquad \textbf{(B.11)}$$

Note that since **c** does not appear in **A** for the linear SWE, this is the same matrix as the Roe linearized matrix in *Leveque* [1990], and it is not necessary to compute the Roe averaged variables. Next the eigenvalues and eigenvectors of **A** are calculated to obtain

$$\lambda_1 = -a \; , \; \lambda_2 = a \; , \; \mathbf{v}_1 = \begin{bmatrix} 1 \\ -a/h \end{bmatrix} \text{ , and } \mathbf{v}_2 = \begin{bmatrix} 1 \\ a/h \end{bmatrix} \qquad \textbf{(B.12)}$$

where $a = \sqrt{gh}$ is the frictionless wave celerity, or the speed at which the wave would travel analytically in the absence of friction.

Now the Riemann problem can be solved using Roe's approximation, which relates the variable states at a shock. For linear hyperbolic systems, an initial state that is discontinuous across an interface will evolve in time such that the discontinuities propagate with the characteristic speeds given by the eigenvalues. Figure B.3 illustrates the evolution of the Riemann problem in time.

These three states are related by the following equations:

$$\mathbf{c}_m = \mathbf{c}_L + \alpha_1 \mathbf{v}_1 \qquad \textbf{(B.13)}$$

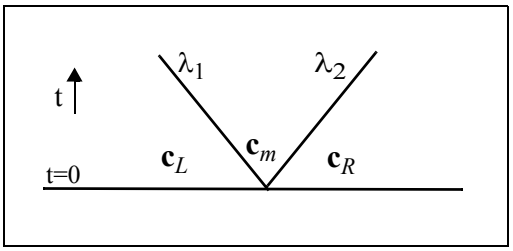


Figure B.3 Evolution of the discontinuity in time.

$$\mathbf{c}_R = \mathbf{c}_m + \alpha_2 \mathbf{v}_2 = \mathbf{c}_L + \alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2 , \tag{B.14}$$

which are simply a linear combination of the eigenvectors. Now use Equation (B.14) to solve for α_1 and α_2 in terms of the state variables, ζ and u:

$$\begin{bmatrix} \zeta_R \\ u_R \end{bmatrix} = \begin{bmatrix} \zeta_L \\ u_L \end{bmatrix} + \alpha_1 \begin{bmatrix} 1 \\ -a/h \end{bmatrix} + \alpha_2 \begin{bmatrix} 1 \\ a/h \end{bmatrix} \tag{B.15}$$

The first equation can be used to solve for α_1 in terms of the variables and α_2

$$\zeta_R = \zeta_L + \alpha_1 + \alpha_2 \quad \Rightarrow \quad \alpha_1 = \zeta_R - \zeta_L - \alpha_2 \tag{B.16}$$

then substitute α_1 into the second equation to get

$$u_R = u_L + \frac{a}{h}(- \alpha_1 + \alpha_2) = u_L + \frac{a}{h}(-\zeta_R + \zeta_L + \alpha_2 + \alpha_2), \tag{B.17}$$

which gives α_2 as

$$\frac{h}{a}(u_R - u_L) + \zeta_R - \zeta_L = 2\alpha_2 \tag{B.18}$$

so that the final values can be written as

$$\alpha_1 = \frac{1}{2}\left[(\zeta_R - \zeta_L) - \frac{h}{a}(u_R - u_L)\right] \text{ and} \tag{B.19}$$

$$\alpha_2 = \frac{1}{2}\left[(\zeta_R - \zeta_L) + \frac{h}{a}(u_R - u_L)\right]. \tag{B.20}$$

Then the Roe numerical flux, using the left state and summing over the p eigenstates, is given by

$$\mathbf{F}(\mathbf{c}_L, \mathbf{c}_R) = \mathbf{f}_L + \sum_p \lambda_p^- \alpha_p \mathbf{v}_p \text{ where } \lambda_p^- = min[\lambda_p, 0] \quad \textbf{(B.21)}$$

or using the right state

$$\mathbf{F}(\mathbf{c}_L, \mathbf{c}_R) = \mathbf{f}_R - \sum_p \lambda_p^- \alpha_p \mathbf{v}_p \text{ where } \lambda_p^- = max[\lambda_p, 0] \; . \quad \textbf{(B.22)}$$

For this study, the left state is used and the numerical flux is computed as

$$\mathbf{F} = \mathbf{f}_L + \sum_p \lambda_p^- \alpha_p \mathbf{v}_p = \mathbf{f}_L + \lambda_1^- \alpha_1 \mathbf{v}_1 + \lambda_2^- \alpha_2 \mathbf{v}_2 = \mathbf{f}_L + \lambda_1 \alpha_1 \mathbf{v}_1 \quad \textbf{(B.23)}$$

since $\lambda_2^- = min[a, 0] = 0$.

Substituting the definition for **f** and Equations (B.12) and (B.19) gives the numerical flux

$$\mathbf{F} = \begin{bmatrix} hu_L \\ g\zeta_L \end{bmatrix} - \frac{a}{2}\left((\zeta_R - \zeta_L) - \frac{h}{a}(u_R - u_L)\right)\begin{bmatrix} 1 \\ -a/h \end{bmatrix} , \quad \textbf{(B.24)}$$

which can be simplified to give

$$\mathbf{F} = \frac{1}{2}\begin{bmatrix} h(u_L + u_R) - a(\zeta_R - \zeta_L) \\ g(\zeta_L + \zeta_R) - a(u_R - u_L) \end{bmatrix} . \quad \textbf{(B.25)}$$

Now return to Equation (B.9) and substitute the numerical flux **F** for the boundary fluxes $\mathbf{f}_L$ and $\mathbf{f}_R$ to arrive at the general form of the discrete FVM equations

$$\zeta_j^{k+1} = \zeta_j^k - \frac{\Delta t}{2\Delta x}[h(u_{j,R}^k + u_{j+1,L}^k - u_{j-1,R}^k - u_{j,L}^k) -$$

$$a(\zeta_{j-1,R}^k - \zeta_{j,L}^k - \zeta_{j,R}^k + \zeta_{j+1,L}^k)] \quad \textbf{(B.26)}$$

$$u_j^{k+1} = u_j^k - \frac{\Delta t}{2\Delta x}[g(\zeta_{j,R}^k + \zeta_{j+1,L}^k - \zeta_{j-1,R}^k - \zeta_{j,L}^k)$$

$$-a(u^k_{j-1,R} - u^k_{j,L} - u^k_{j,R} + u^k_{j+1,L})] - \tau u^k_j \Delta t\,, \qquad \textbf{(B.27)}$$

where again a is the frictionless wave celerity (or the speed at which the wave would travel analytically in the absence of friction) and is equal to $\sqrt{gh}$, the grid spacing Δx is assumed to be constant, and the subscripts $_{j,\,R(L)}$ indicate the rightmost (leftmost) value of u or ζ within the j^{th} cell. These values are calculated from the known cell-centered average and the assumed functional form of the interpolates within the cell. Herein, only piecewise constant and piecewise linear approximations are considered, where the raw slopes are calculated using values from neighboring cells (explicit slope reconstruction after the computation of the cell average quantities). These general equations are valid for both low- and high-order FVM approximations, since the assumed form of the interpolates do not enter in until the left and right cell values are determined.

Looking at Equations (B.26) and (B.27), and noting that for piecewise constants $u_{j,L} = u_{j,R}$ and $\zeta_{j,\,L} = \zeta_{j,\,R}$, the low-order discrete FVM representation can be written as

$$\zeta^{k+1}_j = \zeta^k_j - \frac{\Delta t}{2\Delta x}[h(u^k_{j+1} - u^k_{j-1}) - a(\zeta^k_{j-1} - 2\zeta^k_j + \zeta^k_{j+1})] \qquad \textbf{(B.28)}$$

$$u^{k+1}_j = (1 - \tau\Delta t)u^k_j - \frac{\Delta t}{2\Delta x}[g(\zeta^k_{j+1} - \zeta^k_{j-1}) - a(u^k_{j-1} - 2u^k_j + u^k_{j+1})]\,. \qquad \textbf{(B.29)}$$

For stability, some limiting procedure must be applied to the raw slopes when linear interpolates are used; this is so that two adjacent cells do not have slopes of opposite sign. (Refer to Figure B.2 for a discretization where the linear interpolates have been limited. Notice that the j^{th} cell has a slope of zero.) These limiting procedures are also known as total-variation diminishing (TVD) and essentially non-oscillatory (ENO) schemes. They prohibit expressing the high-order FVM algorithm in a completely closed-form equation, since the slopes evolve with the solution. For this study, three common limiters (as presented in various applications of higher-order methods) and their behavior within the

context of the high-order FVM discretization of the SWE have been examined. These are the minmod limiter used by *Sweby* [1984], given as

$$\text{minmod slope} = s_L \text{Min}(|L|, |R|) \quad \text{if } s_L s_R > 0, \quad \text{else slope} = 0, \tag{B.30}$$

the Superbee limiter used by *Causon et al.* [2000], given as

$$\text{Superbee slope} = s_L \text{Max}[0, \text{Min}(2|L|, s_L|R|), \text{Min}(|L|, 2s_L|R|)], \tag{B.31}$$

and the vanLeer limiter used by *Bell and Shubin* [1984], given as

$$\text{vanLeer slope} = 0.5 s_C \text{Min}(0.5C, \text{Minmod}(L, R)), \tag{B.32}$$

where s_x is the sign of x and the raw slopes are calculated using standard finite differences:

$$L = \frac{b_j - b_{j-1}}{\Delta x}, \; R = \frac{b_{j+1} - b_j}{\Delta x}, \text{ and } C = \frac{b_{j+1} - b_{j-1}}{2\Delta x} \tag{B.33}$$

where b can be either u or ζ, the subscripts indicate the cell average values for the j^{th} cell, and the grid spacing Δx is assumed constant. Although mathematically each of these limiters takes on a slightly different form, they are essentially performing the same tasks: 1) to ensure a monotonic increase or decrease in the function values by prohibiting slopes of opposite sign in adjacent cells (i.e., $s_L s_R > 0$ tests that slopes have the same sign) and 2) to limit the magnitude of the boundary cell's slope to prevent overshoot and undershoot of the boundary conditions.

B.4 Discontinuous Galerkin finite element

Discontinuous Galerkin finite element methods are similar to traditional Galerkin FE methods in that both solve the weak form of the equations by multiplying by a weight function and integrating. As in traditional Galerkin approximations, the weight functions are identical to the interpolating functions. There are, however, two crucial differences

between continuous Galerkin (CG) approximations and discontinuous approximations. The first distinction is that DG methods allow discontinuities at the element interfaces, which in effect disconnects adjacent elements and allows each to be solved independently from the others. (As will be seen, each element does require information from adjoining elements, but this information enters in explicitly (for explicit time stepping) and does not couple the elements.) The second distinction follows from the first in that DG methods integrate over a single element using local weight and interpolating functions, while continuous methods integrate over the entire domain and use global weight functions, which are zero outside of the element of interest. These two differences provide certain advantages to the DG method including: local *p*-refinement (i.e., each element has its own basis and weight function allowing higher order interpolates in known problem areas), local *h*-refinement (i.e., each element is not required to match up with its neighbors and hanging nodes do not pose a problem), ability to capture discontinuous solutions and handle shocks, and local mass conservation since equations are satisfied on the element level rather than globally.

Additionally, the DG method can be thought of as a generalization of the FVM. Although both solve the equations in a weak sense, they differ in how they apply the interpolates. FVM schemes use explicit post-processing to incorporate the higher-order terms, while DG methods assume the form of the interpolate from the beginning and use finite element techniques; both the higher-order DG and FVM can require slope limiting for stability. Herein piecewise constant (low-order) and piecewise linear (high-order) approximations will be derived. It will be shown that the low-order FVM and low-order DG algorithms are equivalent. The grid structure of the DG method is the same as the FVM with unknowns centered on an element; see Figure B.2.

The starting point for the DG discretization of the shallow water equations is the "local" weak form of the equations. Multiply each equation by a weight function, $\phi_i(x)$, and

integrate by parts on an element to get

$$(\frac{\partial \zeta}{\partial t},\phi_i)_{\Omega_e} - (hu,\frac{\partial \phi_i}{\partial x})_{\Omega_e} + \langle hu \cdot n_e, \phi_i^- \rangle_{\partial\Omega_e} = 0 \text{ and} \tag{B.34}$$

$$(\frac{\partial u}{\partial t},\phi_i)_{\Omega_e} - (g\zeta,\frac{\partial \phi_i}{\partial x})_{\Omega_e} + \langle g\zeta \cdot n_e, \phi_i^- \rangle_{\partial\Omega_e} + (\tau u,\phi_i)_{\Omega_e} = 0\,, \tag{B.35}$$

where n_e is the unit outward normal, $\partial\Omega_e$ is the boundary of element *e*, and ϕ_i^- indicates that the weight function is evaluated internal to the element at the boundary discontinuity.

The allowed discontinuities in ζ and *u* become a problem when you try to evaluate the boundary fluxes (*hu* and $g\zeta$), and an appropriate numerical flux, **F**, must be substituted for these terms. This requires solving the Riemann problem at each element interface. See §B.3 for a complete derivation of Roe's approximation applied to the shallow water equations, where it was shown in Equation (B.25) that the appropriate numerical fluxes are

$$\mathbf{F} = \frac{1}{2}\begin{bmatrix} h(u_L + u_R) - a(\zeta_R - \zeta_L) \\ g(\zeta_L + \zeta_R) - a(u_R - u_L) \end{bmatrix} \tag{B.36}$$

where $a = \sqrt{gh}$ is the wave celerity, and the subscripts $_L$ and $_R$ denote the left and right states at the boundary. Notice that these approximations add an additional term to the simple arithmetic average of the left and right flux terms (*hu* and $g\zeta$).

The next step is to approximate the dependent variables ζ and *u* with some basis function. Herein piecewise constant and piecewise linear Legendre bases are examined. Although Lagrange bases can be more intuitive to work with, since the unknowns are located at the node points, as in continuous schemes, rather than at the element center, Legendre bases are orthogonal and more easily accommodate higher-order terms, as higher terms are simply added to the existing approximation rather than requiring a complete rederivation of the bases. The fully discrete DG equations will be worked out individually

for each of these bases in the following subsections.

Piecewise constant

The structure of an element with piecewise constant interpolates is shown below in Figure B.4. Note that there is only one unknown per element and the values at the element boundaries are simply equal to the element average value at the center.

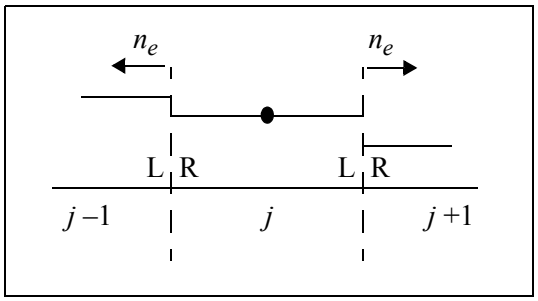


Figure B.4 Structure of element j with piecewise constant basis functions.

The approximations on an element are given as

$$\zeta^h = \zeta_j \phi_0 \text{ and } u^h = u_j \phi_0 \qquad \textbf{(B.37)}$$

where the basis function is simply $\phi_0 = 1.0$. Now substitute ζ^h and u^h for the state variables and the approximate flux for the boundary terms in Equations (B.34) and (B.35) to arrive at

$$\frac{\partial}{\partial t}(\zeta_j \phi_0, \phi_0)_{\Omega_e} - (h u_j \phi_0, \frac{\partial \phi_0}{\partial x})_{\Omega_e} + \frac{1}{2}[h(u_L + u_R) - a(\zeta_R - \zeta_L)] \cdot n_e \phi_0^- \Big|_{x_{j-1/2}}^{x_{j+1/2}} = 0 \qquad \textbf{(B.38)}$$

$$\frac{\partial}{\partial t}(u_j \phi_0, \phi_0)_{\Omega_e} - (g \zeta \phi_0, \frac{\partial \phi_0}{\partial x})_{\Omega_e} + (\tau u_j \phi_0, \phi_0)_{\Omega_e} +$$

$$\cdot\frac{1}{2}[g(\zeta_L+\zeta_R)-a(u_R-u_L)]\cdot n_e\phi_0^- \Big|_{x_{j-1/2}}^{x_{j+1/2}} = 0\,. \qquad \textbf{(B.39)}$$

The second integral term in each equation vanishes since $\frac{\partial}{\partial x}1 = 0$ and the first integral evaluates simply to $x_{j+1/2} - x_{j-1/2} = \Delta x_j$. Also, note that the interface values are simply $b_{j-1/2,L} = b_{j-1}$, $b_{j-1/2,R} = b_j$, $b_{j+1/2,L} = b_j$, and $b_{j+1/2,R} = b_{j+1}$, where b is either ζ or u, to get after simplification

$$\frac{\partial}{\partial t}\Delta x_j\zeta_j+\frac{1}{2}[h(u_{j-1}+2u_j+u_{j+1})-a(\zeta_{j+1}-\zeta_{j-1})] = 0 \qquad \textbf{(B.40)}$$

$$\frac{\partial}{\partial t}\Delta x_j u_j+\frac{1}{2}[g(\zeta_{j-1}+2\zeta_j+\zeta_{j+1})-a(u_{j+1}-u_{j-1})]+\Delta x_j\tau u_j = 0\,. \qquad \textbf{(B.41)}$$

With an explicit temporal discretization, the low-order DG algorithm for the linearized SWE is given as:

$$\zeta_j^{k+1} = \zeta_j^k+\frac{\Delta t}{2\Delta x}[h(u_{j-1}^k+2u_j^k+u_{j+1}^k)-a(\zeta_{j+1}^k-\zeta_{j-1}^k)] \text{ and} \qquad \textbf{(B.42)}$$

$$u_j^{k+1} = (1-\tau\Delta t)u_j^k+\frac{\Delta t}{2\Delta x}[g(\zeta_{j-1}^j+2\zeta_j^k+\zeta_{j+1}^k)-a(u_{j+1}^k-u_{j-1}^k)]\,. \qquad \textbf{(B.43)}$$

Looking back at Equations (B.28) and (B.29), these are exactly the same equations as were derived for the low-order FVM using piecewise constant data. *Cockburn et al.* [2002] note in their DG history that the DG method can be thought of as a generalization of the FVM.

Piecewise-linear Legendre

The structure of an element in the Legendre basis is shown below in Figure B.5. Note that the variables are evaluated at the element center, such that the unknown values for the interpolate in element *j* are the elevation and velocity element averages and "slopes", which are indicated in the figure with a solid dot.

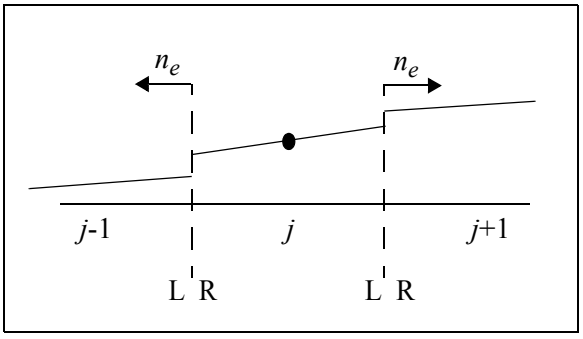


Figure B.5 Structure of element j for piecewise linear Legendre basis functions.

The approximations in this basis are written as

$$\zeta^h = \zeta_j^{(0)}\phi_0 + \zeta_j^{(1)}\phi_1 \text{ and } u^h = u_j^{(0)}\phi_0 + u_j^{(1)}\phi_1 \qquad \textbf{(B.44)}$$

where $\zeta(u)_j^{(0)}$ and $\zeta(u)_j^{(1)}$ are the element average and "slope" respectively, and ϕ_0 and ϕ_1 are the standard linear Legendre interpolates presented in numerical methods texts:

$$\phi_0 = 1 \text{ and } \phi_1 = \frac{12}{\Delta x_j}(x - x_j) \ . \qquad \textbf{(B.45)}$$

Note that the factor $12/\Delta x_j$ is required to normalize the basis (see *Cockburn and Shu* [1989] for more details), such that the actual slope in the element is given as,

$$\text{slope} = \frac{12}{\Delta x_j}\zeta(u)_j^{(1)} . \qquad \textbf{(B.46)}$$

Now substitute ζ^h and u^h for the state variables and the numerical flux **F** into Equations (B.34) and (B.35) to arrive at

$$\frac{\partial}{\partial t}(\zeta_j^{(0)}\phi_0 + \zeta_j^{(1)}\phi_1, \phi_i) - h(u_j^{(0)}\phi_0 + u_j^{(1)}\phi_1, \frac{\partial \phi_i}{\partial x}) +$$

$$\frac{1}{2}[h(u_L+u_R)-a(\zeta_R-\zeta_L)]\cdot n_e\phi_i^- \Big|_{x_{j-1/2}}^{x_{j+1/2}} = 0 \qquad \textbf{(B.47)}$$

$$\frac{\partial}{\partial t}(u_j^{(0)}\phi_0+u_j^{(1)}\phi_1,\phi_i)_{\Omega_e}-g(\zeta^{(0)}\phi_0+\zeta^{(1)}\phi_1,\frac{\partial\phi_i}{\partial x})_{\Omega_e}+\tau(u_j\phi_0+u_j^{(1)}\phi_1,\phi_i)_{\Omega_e}+$$

$$\frac{1}{2}[g(\zeta_L+\zeta_R)-a(u_R-u_L)]\cdot n_e\phi_i^- \Big|_{x_{j-1/2}}^{x_{j+1/2}} = 0 \qquad \textbf{(B.48)}$$

Integrate with each basis function, ϕ_i , successively to generate 4 equations for the 4 unknowns ($\zeta_j^{(0)}$, $\zeta_j^{(1)}$, $u_j^{(0)}$, $u_j^{(1)}$). Evaluation of the integral combinations gives the following:

$$\int_{x_{j-1/2}}^{x_{j+1/2}}\phi_0\phi_0 dx = (x_{j+1/2}-x_{j-1/2}) = \Delta x_j$$

$$\int_{x_{j-1/2}}^{x_{j+1/2}}\phi_0\phi_1 dx = \int_{x_{j-1/2}}^{x_{j+1/2}}\phi_1\phi_0 dx = 0$$

$$\int_{x_{j-1/2}}^{x_{j+1/2}}\phi_1\phi_1 dx = 12\Delta x_j$$

$$\int_{x_{j-1/2}}^{x_{j+1/2}}\phi_0\frac{\partial\phi_0}{\partial x}dx = \int_{x_{j-1/2}}^{x_{j+1/2}}\phi_1\frac{\partial\phi_0}{\partial x}dx = \int_{x_{j-1/2}}^{x_{j+1/2}}\phi_1\frac{\partial\phi_1}{\partial x}dx = 0$$

$$\int_{x_{j-1/2}}^{x_{j+1/2}}\phi_0\frac{\partial\phi_1}{\partial x}dx = 12$$

and substitution of these into the integral equations results (after some simplification) in the following four equations:

<u>i = 0</u>

$$\Delta x_j\frac{\partial}{\partial t}\zeta_j^{(0)} = \frac{1}{2}[h(u_L+u_R)-a(\zeta_R-\zeta_L)]\Big|_{j-1/2}-\frac{1}{2}[h(u_L+u_R)-a(\zeta_R-\zeta_L)]\Big|_{j+1/2} \qquad \textbf{(B.49)}$$

$$\Delta x_j\left(\frac{\partial}{\partial t}+\tau\right)u_j^{(0)} = \frac{1}{2}[g(\zeta_L+\zeta_R)-a(u_R-u_L)]\Big|_{j-1/2} -$$

$$\frac{1}{2}[g(\zeta_L+\zeta_R)-a(u_R-u_L)]\Big|_{j+1/2} \qquad \textbf{(B.50)}$$

<u>i = 1</u>

$$\Delta x_j\frac{\partial}{\partial t}\zeta_j^{(1)} = hu_j^{(0)} - \frac{1}{4}[h(u_L+u_R)-a(\zeta_R-\zeta_L)]\Big|_{j-1/2} -$$

$$\frac{1}{4}[h(u_L+u_R)-a(\zeta_R-\zeta_L)]\Big|_{j+1/2} \qquad \textbf{(B.51)}$$

$$\Delta x_j\left(\frac{\partial}{\partial t}+\tau\right)u_j^{(1)} = g\zeta_j^{(0)} - \frac{1}{4}[g(\zeta_L+\zeta_R)-a(u_R-u_L)]\Big|_{j-1/2} -$$

$$\frac{1}{4}[g(\zeta_L+\zeta_R)-a(u_R-u_L)]\Big|_{j+1/2} \qquad \textbf{(B.52)}$$

where the boundary terms need to be evaluated at j–1/2 and j+1/2 and the basis functions at the boundary have already been evaluated.

The evaluations at the boundary require that the form of the basis functions be used to write the boundary values in terms of the element average and slope. From the approximations, the following are derived:

$$u_{j-1/2_L} = u_{j-1}^{(0)} + 6u_{j-1}^{(1)}$$

$$u_{j-1/2_R} = u_j^{(0)} - 6u_j^{(1)}$$

$$u_{j+1/2_L} = u_j^{(0)} + 6u_j^{(1)}$$

$$u_{j+1/2_R} = u_{j+1}^{(0)} - 6u_{j+1}^{(1)}$$

where the same expressions can also be written for the elevation, ζ. Substitution of these expressions into Equations (B.49) through (B.52) and use of an explicit temporal

discretization gives the high-order DG algorithm for the linearized SWE as:

$$\zeta_j^{(0)k+1} = \zeta_j^{(0)k} + \frac{\Delta t}{2\Delta x_j}[h(u_{j-1}^{(0)k} + 6u_{j-1}^{(1)k} + u_j^{(0)k} - 6u_j^{(1)k}) -$$

$$a(\zeta_j^{(0)k} - 6\zeta_j^{(1)k} - (\zeta_{j-1}^{(0)k} + 6\zeta_{j-1}^{(1)k}))] - \frac{\Delta t}{2\Delta x_j}[h(u_j^{(0)k} + 6u_j^{(1)k} + u_{j+1}^{(0)k} - 6u_{j+1}^{(1)k}) -$$

$$a(\zeta_{j+1}^{(0)k} - 6\zeta_{j+1}^{(1)k} - (\zeta_j^{(0)k} + 6\zeta_j^{(1)k}))] \quad \textbf{(B.53)}$$

$$u_j^{(0)k+1} = (1-\tau\Delta t)u_j^{(0)k} + \frac{\Delta t}{2\Delta x_j}[g(\zeta_{j-1}^{(0)k} + 6\zeta_{j-1}^{(1)k} + \zeta_j^{(0)k} - 6\zeta_j^{(1)k}) -$$

$$a(u_j^{(0)k} - 6u_j^{(1)k} - (u_{j-1}^{(0)k} + 6u_{j-1}^{(1)k}))] - \frac{\Delta t}{2\Delta x_j}[g(\zeta_j^{(0)k} + 6\zeta_j^{(1)k} + \zeta_{j+1}^{(0)k} - 6\zeta_{j+1}^{(1)k}) -$$

$$a(u_{j+1}^{(0)k} - 6u_{j+1}^{(1)k} - (u_j^{(0)k} + 6u_j^{(1)k}))] \quad \textbf{(B.54)}$$

$$\zeta_j^{(1)k+1} = \zeta_j^{(1)k} + \frac{\Delta t}{\Delta x_j}hu_j^{(0)k} - \frac{\Delta t}{4\Delta x_j}[h(u_{j-1}^{(0)k} + 6u_{j-1}^{(1)k} + u_j^{(0)k} - 6u_j^{(1)k}) -$$

$$a(\zeta_j^{(0)k} - 6\zeta_j^{(1)k} - (\zeta_{j-1}^{(0)k} + 6\zeta_{j-1}^{(1)k}))] - \frac{\Delta t}{4\Delta x_j}[h(u_j^{(0)k} + 6u_j^{(1)k} + u_{j+1}^{(0)k} - 6u_{j+1}^{(1)k}) -$$

$$a(\zeta_{j+1}^{(0)k} - 6\zeta_{j+1}^{(1)k} - (\zeta_j^{(0)k} + 6\zeta_j^{(1)k}))] \quad \textbf{(B.55)}$$

$$u_j^{(1)k+1} = (1-\tau\Delta t)u_j^{(1)k} + \frac{\Delta t}{\Delta x_j}g\zeta_j^{(0)k} - \frac{\Delta t}{4\Delta x_j}[g(\zeta_{j-1}^{(0)k} + 6\zeta_{j-1}^{(1)k} + \zeta_j^{(0)k} - 6\zeta_j^{(1)k}) -$$

$$a(u_j^{(0)k} - 6u_j^{(1)k} - (u_{j-1}^{(0)k} + 6u_{j-1}^{(1)k}))] - \frac{\Delta t}{4\Delta x_j}[g(\zeta_j^{(0)k} + 6\zeta_j^{(1)k} + \zeta_{j+1}^{(0)k} - 6\zeta_{j+1}^{(1)k}) -$$

$$a(u_{j+1}^{(0)k} - 6u_{j+1}^{(1)k} - (u_j^{(0)k} + 6u_j^{(1)k}))] \ , \quad \textbf{(B.56)}$$

where the parenthetical superscripts (0) and (1) represent the two degrees of freedom (DOF) for each state variable. The zero DOF is the element average for the variable and the one DOF is the slope within the element. The *k* superscripts are from the temporal discretization and indicate at which time level the terms should be evaluated. Notice that

although the number of unknowns has only doubled relative to the low-order DG (4 versus 2), the number of computations has more than doubled.

As in the high-order FVM, slope limiters may be necessary for added stability; however, of the limiters that were used in the FVM, only the minmod limiter will be applied to the high-order DG in this study. (The minmod limiter with a total variation bounded (TVB) correction was used in the original development of the Runge-Kutta DG method by *Cockburn and Shu* [1991].) Additionally, some changes are necessary relative to the limiting procedure used for the FVM: 1) the limiter must compare the calculated DG slopes to the left and right slopes and 2) the centered slopes C are taken to be the DG state values times the normalization factor, $(12/\Delta x_j)\zeta_j^{(1)}$ and $(12/\Delta x_j)u_j^{(1)}$, rather than calculating them as a centered difference. With these changes, the limiter in the DG framework is given as

$$\text{minmod slope} = s_C \text{Min}(|L|, |C|, |R|) \quad \text{if } s_L = s_C = s_R \text{ , else slope} = 0 \qquad \textbf{(B.57)}$$

where the left and right slopes are calculated using the element averages from adjacent elements, as in the FVM (see Equation). For the DG algorithm, slope limiting is applied as a post-processing step; that is, after the new slope values have been computed they are limited, and the calculated values replaced by the limited values, in preparation for the next time step.

For non-constant grid spacing, the limiter is expressed as given in Equation (B.57); but the slopes are calculated as

$$L = \frac{b_j^{(0)} - b_{j-1}^{(0)}}{0.5(\Delta x_j + \Delta x_{j-1})}, \; C = \frac{12}{\Delta x_j} b_j^{(1)}, \text{ and } R = \frac{b_{j+1}^{(0)} - b_j^{(0)}}{0.5(\Delta x_{j+1} + \Delta x_j)} \qquad \textbf{(B.58)}$$

where b can be either state variable, ζ or u. After the slopes are limited, the new value for the (1) degree of freedom is given as

$$b_j^{(1)} = \frac{\Delta x_j}{12} \text{slope}. \quad \textbf{(B.59)}$$

APPENDIX C

Derivation of Analytical Fourier and Dispersion Relationships for the SWE Study Algorithms

C.1 Fourier analysis

C.1.1 Theory

The Fourier analysis for a general discretization of the SWE is presented herein, for more detail about the theory see Chapter 2. For a fully discretized system, a single component of the Fourier series, represented as

$$b_j^{\,k} = b_0 \mathrm{e}^{i\omega k\Delta t} \mathrm{e}^{i\sigma j\Delta x}, \qquad \textbf{(C.1)}$$

where j, k are the spatial and temporal discretization indices, respectively, b can represent either ζ or u, and b_0 is the amplitude, can be substituted into the discrete representation of Equations (1.5), (1.6), and (1.7) for each independent variable. The discrete per-time-step propagation factor is defined as:

$$\lambda' \equiv \frac{b_j^{\,k+1}}{b_j^{\,k}} = \mathrm{e}^{i\omega\Delta t} \qquad \textbf{(C.2)}$$

where the prime differentiates the discrete propagation factor from the continuum factor. Definitions (C.1) and (C.2) are substituted into the set of discrete equations for the

algorithm under study and a new system results;

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} \zeta_0 \\ u_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{C.3}$$

where the form of the matrix coefficients will vary depending upon the spatial and temporal discretization. A nontrivial solution will exist only when the determinant of the coefficient matrix is zero, which results in a polynomial in the discrete per-time-step propagation factor λ', whose complex roots can then be written as a function of the wave number, σ.

In the following sections, the Fourier propagation factor will be derived for each of the SWE study algorithms. A complete derivation will be given for the first algorithm (SFD) to illustrate the procedure, after which a more abbreviated derivation will be used for the remaining algorithms.

C.1.2 Staggered finite difference

The discrete equations for the staggered finite difference approximation to the SWE using weighted Euler time stepping are given as

$$\zeta_j^{k+1} + h\theta\frac{\Delta t}{\Delta x}(u_{j+1/2}^{k+1} - u_{j-1/2}^{k+1}) = \zeta_j^k - h(1-\theta)\frac{\Delta t}{\Delta x}(u_{j+1/2}^k - u_{j-1/2}^k) \tag{C.4}$$

$$(1+\tau\Delta t\theta)u_{j+1/2}^{k+1} + g\theta\frac{\Delta t}{\Delta x}(\zeta_{j+1}^{k+1} - \zeta_j^{k+1}) =$$

$$(1-\tau\Delta t(1-\theta))u_{j+1/2}^k - g(1-\theta)\frac{\Delta t}{\Delta x}(\zeta_{j+1}^k - \zeta_j^k) \tag{C.5}$$

where θ is the time weighting parameter, which varies between 0 (explicit) and 1 (implicit), and the equations are as derived in Appendix B.

Now substitute a single Fourier component for each variable, given as

$$\zeta_j^{\,k} = \zeta_0 e^{i\omega k\Delta t} e^{i\sigma j\Delta x} \text{ and } u_j^{\,k} = u_0 e^{i\omega k\Delta t} e^{i\sigma j\Delta x}, \tag{C.6}$$

into the discrete equations to get

$$\zeta_0 e^{i\omega(k+1)\Delta t} e^{i\sigma j\Delta x} + h\theta\frac{\Delta t}{\Delta x}(u_0 e^{i\omega(k+1)\Delta t} e^{i\sigma(j+1/2)\Delta x} - u_0 e^{i\omega(k+1)\Delta t} e^{i\sigma(j-1/2)\Delta x}) =$$

$$\zeta_0 e^{i\omega k\Delta t} e^{i\sigma j\Delta x} - h(1-\theta)\frac{\Delta t}{\Delta x}(u_0 e^{i\omega k\Delta t} e^{i\sigma(j+1/2)\Delta x} - u_0 e^{i\omega k\Delta t} e^{i\sigma(j-1/2)\Delta x}) \tag{C.7}$$

$$(1+\tau\Delta t\theta)u_0 e^{i\omega(k+1)\Delta t} e^{i\sigma(j+1/2)\Delta x} + g\theta\frac{\Delta t}{\Delta x}(\zeta_0 e^{i\omega(k+1)\Delta t} e^{i\sigma(j+1)\Delta x} -$$

$$\zeta_0 e^{i\omega(k+1)\Delta t} e^{i\sigma j\Delta x}) = (1-\tau\Delta t(1-\theta))u_0 e^{i\omega k\Delta t} e^{i\sigma(j+1/2)\Delta x} -$$

$$g(1-\theta)\frac{\Delta t}{\Delta x}(\zeta_0 e^{i\omega k\Delta t} e^{i\sigma(j+1)\Delta x} - \zeta_0 e^{i\omega k\Delta t} e^{i\sigma j\Delta x}). \tag{C.8}$$

Next divide through by the common factor, $e^{i\omega k\Delta t} e^{i\sigma j\Delta x}$, in each equation and gather like terms to get

$$\zeta_0 e^{i\omega\Delta t} + h\theta\frac{\Delta t}{\Delta x}u_0 e^{i\omega\Delta t}(e^{i\sigma\Delta x/2} - e^{-i\sigma\Delta x/2}) =$$

$$\zeta_0 - h(1-\theta)\frac{\Delta t}{\Delta x}u_0(e^{i\sigma\Delta x/2} - e^{-i\sigma\Delta x/2}) \tag{C.9}$$

$$(1+\tau\Delta t\theta)u_0 e^{i\omega\Delta t} e^{i\sigma\Delta x/2} + g\theta\frac{\Delta t}{\Delta x}\zeta_0 e^{i\omega\Delta t}(e^{i\sigma\Delta x} - 1) =$$

$$(1-\tau\Delta t(1-\theta))u_0 e^{i\sigma\Delta x/2} - g(1-\theta)\frac{\Delta t}{\Delta x}\zeta_0(e^{i\sigma\Delta x} - 1). \tag{C.10}$$

Now note the definition for the propagation factor given in Equation (C.2) as $\lambda' = e^{i\omega\Delta t}$, define $\mu = \Delta t/\Delta x$ and use the identity

$$e^{\pm i\sigma x} = \cos\sigma x \pm i\sin\sigma x \tag{C.11}$$

to get

$$\zeta_0\lambda' + h\theta\mu\lambda' 2i\sin(\sigma\Delta x/2)u_0 = \zeta_0 - h(1-\theta)\mu 2i\sin(\sigma\Delta x/2)u_0 \quad \textbf{(C.12)}$$

$$(1+\tau\Delta t\theta)\lambda' e^{i\sigma\Delta x/2}u_0 + g\theta\mu\lambda'(e^{i\sigma\Delta x}-1)\zeta_0 =$$

$$(1-\tau\Delta t(1-\theta))e^{i\sigma\Delta x/2}u_0 - g(1-\theta)\mu(e^{i\sigma\Delta x}-1)\zeta_0 . \quad \textbf{(C.13)}$$

Then write the equations as a system for ζ_0, u_0 to get

$$\begin{bmatrix} (\lambda'-1) & 2i\sin(\sigma\Delta x/2)h\mu(\theta\lambda'+1-\theta) \\ g\mu(e^{i\sigma\Delta x}-1)(\theta\lambda'+1-\theta) & e^{i\sigma\Delta x/2}(\lambda'-1+\tau\Delta t(\theta\lambda'+1-\theta)) \end{bmatrix} \begin{bmatrix} \zeta_0 \\ u_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \textbf{(C.14)}$$

and set the determinant equal to zero to get a polynomial expression for λ':

$$(\lambda'-1)e^{i\sigma\Delta x/2}(\lambda'-1+\tau\Delta t(\theta\lambda'+1-\theta)) -$$

$$2i\sin(\sigma\Delta x/2)gh\mu^2(\theta\lambda'+1-\theta)^2(e^{i\sigma\Delta x}-1) = 0, \quad \textbf{(C.15)}$$

which gives, after gathering like powers of λ'

$$[e^{i\sigma\Delta x/2}(1+\tau\Delta t\theta) - (e^{i\sigma\Delta x}-1)2i\sin(\sigma\Delta x/2)gh\mu^2\theta^2]\lambda'^2 +$$

$$[e^{i\sigma\Delta x/2}(-2+\tau\Delta t(1-2\theta)) - gh\mu^2(e^{i\sigma\Delta x}-1)(2i\sin(\sigma\Delta x/2)(2\theta(1-\theta)))]\lambda' +$$

$$[e^{i\sigma\Delta x/2}(1-\tau\Delta t(1-\theta)) - gh\mu^2(e^{i\sigma\Delta x}-1)(2i\sin(\sigma\Delta x/2)(1-\theta)^2)] = 0. \quad \textbf{(C.16)}$$

Now for simplicity, define two new parameters

$$S = (e^{i\sigma\Delta x}-1)2i\sin(\sigma\Delta x/2)gh\mu^2 \text{ and} \quad \textbf{(C.17)}$$

$$T = e^{i\sigma\Delta x/2} \quad \textbf{(C.18)}$$

so that the polynomial expression becomes

$$[T(1+\tau\Delta t\theta)-S\theta^2]\lambda'^2+[T(-2+\tau\Delta t(1-2\theta))-2\theta(1-\theta)S]\lambda'+$$

$$[T(1-\tau\Delta t(1-\theta))-S(1-\theta)^2]\;=\;0\,. \qquad \textbf{(C.19)}$$

Finally, the roots of this quadratic polynomial in λ' can be simplified to give

$$\lambda'\;=\;1+\frac{2S\theta-\tau\Delta t\,T\pm\sqrt{(\tau\Delta t\,T)^2+4ST}}{2(T(1+\tau\Delta t\theta)-S\theta^2)} \qquad \textbf{(C.20)}$$

where the roots are complex conjugates and give the Fourier propagation behavior of the SFD algorithm.

C.1.3 Primitive leap-frog finite element

The discrete equations for the PFE approximation to the SWE using leap-frog time stepping are given as

$$\zeta_{j-1}^{k+1}+4\zeta_j^{k+1}+\zeta_{j+1}^{k+1}\;=\;(\zeta_{j-1}^{k-1}+4\zeta_j^{k-1}+\zeta_{j+1}^{k-1})-6h\frac{\Delta t}{\Delta x}(u_{j+1}^{k}-u_{j-1}^{k}) \qquad \textbf{(C.21)}$$

$$u_{j-1}^{k+1}+4u_j^{k+1}+u_{j+1}^{k+1}\;=\;(1-2\tau\Delta t)(u_{j-1}^{k-1}+4u_j^{k-1}+u_{j+1}^{k-1})-6g\frac{\Delta t}{\Delta x}(\zeta_{j+1}^{k}-\zeta_{j-1}^{k}) \qquad \textbf{(C.22)}$$

where the leap-frog scheme decouples the continuity and momentum equations.

Now substitute a single Fourier component for each variable, given as

$$\zeta_j^{\,k}=\zeta_0 e^{i\omega k\Delta t}e^{i\sigma j\Delta x}\;\text{ and }\;u_j^{\,k}=u_0 e^{i\omega k\Delta t}e^{i\sigma j\Delta x} \qquad \textbf{(C.23)}$$

divide through by the common factor, $e^{i\omega k\Delta t}e^{i\sigma j\Delta x}$, substitute the following propagation factor definitions:

$$\lambda'\;=\;e^{i\omega\Delta t}\;\text{ and }\;\frac{1}{\lambda'}\;=\;e^{-i\omega\Delta t} \qquad \textbf{(C.24)}$$

and gather like terms to get

$$\begin{bmatrix} (A+4)\left(\lambda' - \frac{1}{\lambda'}\right) & 6h\mu B \\ 6g\mu B & (A+4)\left(\lambda' - (1-2\tau\Delta t)\frac{1}{\lambda'}\right) \end{bmatrix} \begin{bmatrix} \zeta_0 \\ u_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \textbf{(C.25)}$$

where

$$A = e^{i\sigma\Delta x} + e^{-i\sigma\Delta x} = 2\cos\sigma\Delta x \qquad \textbf{(C.26)}$$

$$B = e^{i\sigma\Delta x} - e^{-i\sigma\Delta x} = 2i\sin\sigma\Delta x \qquad \textbf{(C.27)}$$

and $\mu = \Delta t/\Delta x$, as defined for the SFD algorithm.

Now set the determinant equal to zero and gather like terms to get

$$[(A+4)^2]\lambda'^2 + [(A+4)^2(-2+2\tau\Delta t) - 36gh\mu^2B^2] +$$

$$[(A+4)^2(1-2\tau\Delta t)]\frac{1}{\lambda'^2} = 0 \qquad \textbf{(C.28)}$$

then multiply by λ'^2 and divide by $(A+4)^2$ to arrive at

$$\lambda'^4 + \left[(-2+2\tau\Delta t) - \frac{36gh\mu^2B^2}{(A+4)^2}\right]\lambda'^2 + [1-2\tau\Delta t] = 0\,. \qquad \textbf{(C.29)}$$

Note that the leap-frog time-stepping scheme results in a quartic polynomial. The roots of this polynomial can be simplified to give the Fourier propagation factor as

$$\lambda' = \sqrt{(1-\tau\Delta t) + \frac{K}{2} \pm \sqrt{\frac{9\mu^2B^2}{(A+4)^2}K - (\tau\Delta t - 1)K + (\tau\Delta t)^2}} \qquad \textbf{(C.30)}$$

where $K = (36gh\mu^2B^2)/(A+4)^2$ and each set of roots are complex conjugates having

the same magnitude and opposite phase.

C.1.4 ADCIRC model (generalized wave continuity)

The discrete equations for the ADCIRC model for the SWE, which uses a three-level temporal scheme centered at level *k* for continuity and a two-level temporal scheme for momentum, are given as

$$\left(\frac{1}{\Delta t^2}+\frac{G}{2\Delta t}\right)\frac{\Delta x}{6}(\zeta_{j-1}^{k+1}+4\zeta_j^{k+1}+\zeta_{j+1}^{k+1})+\frac{\alpha_1 gh}{\Delta x}(-\zeta_{j-1}^{k+1}+2\zeta_j^{k+1}-\zeta_{j+1}^{k+1})\ =$$

$$\frac{2\Delta x}{6\Delta t^2}(\zeta_{j-1}^{k}+4\zeta_j^{k}+\zeta_{j+1}^{k})+\left(-\frac{1}{\Delta t^2}+\frac{G}{2\Delta t}\right)\frac{\Delta x}{6}(\zeta_{j-1}^{k-1}+4\zeta_j^{k-1}+\zeta_{j+1}^{k-1})-$$

$$(G-\tau)\frac{h}{2}(u_{j+1}^{k}-u_{j-1}^{k})-\frac{\alpha_2 gh}{\Delta x}(-\zeta_{j-1}^{k}+2\zeta_j^{k}-\zeta_{j+1}^{k})-$$

$$\frac{\alpha_3 gh}{\Delta x}(-\zeta_{j-1}^{k-1}+2\zeta_j^{k-1}-\zeta_{j+1}^{k-1})+B^k \qquad \textbf{(C.31)}$$

$$\Delta x\left(1+\frac{\tau\Delta t}{2}\right)u_j^{k+1}\ =\ \Delta x\left(1-\frac{\tau\Delta t}{2}\right)u_j^{k}-\frac{g\Delta t}{4}(\zeta_{j+1}^{k+1}+\zeta_{j+1}^{k}-(\zeta_{j-1}^{k+1}+\zeta_{j-1}^{k})) \qquad \textbf{(C.32)}$$

where the temporal weighting parameters, α_n, must satisfy $\alpha_1+\alpha_2+\alpha_3=1.0$ and are usually set equal to 1/3.

Now substitute a single Fourier component for each variable, given as

$$\zeta_j^{\,k}=\ \zeta_0 e^{i\omega k\Delta t}e^{i\sigma j\Delta x}\ \text{and}\ u_j^{\,k}=\ u_0 e^{i\omega k\Delta t}e^{i\sigma j\Delta x}, \qquad \textbf{(C.33)}$$

divide through by the common factor, $e^{i\omega k\Delta t}e^{i\sigma j\Delta x}$, substitute the following propagation factor definitions:

$$\lambda'\ =\ e^{i\omega\Delta t}\ \text{and}\ \frac{1}{\lambda'}\ =\ e^{-i\omega\Delta t} \qquad \textbf{(C.34)}$$

and gather like terms to get

$$\left(\frac{\Delta x}{6\Delta t^2}\left(\lambda' - 2 + \frac{1}{\lambda'}\right) + \frac{G\Delta x}{12\Delta t}\left(\lambda' - \frac{1}{\lambda'}\right)\right)(A+4)\zeta_0 + (G-\tau)\frac{h}{2}Bu_0 +$$

$$\frac{gh}{\Delta x}(2-A)\left(\alpha_1\lambda' + \alpha_2 + \alpha_3\frac{1}{\lambda'}\right)\zeta_0 = 0 \qquad \textbf{(C.35)}$$

$$\left[\frac{\Delta x}{\Delta t}(\lambda' - 1) + \frac{\tau\Delta x}{2}(\lambda' + 1)\right]u_0 + \frac{g}{4}B(\lambda' + 1)\zeta_0 = 0 \qquad \textbf{(C.36)}$$

where A and B are as defined for previous algorithms.

To write this as a system of equations, first define the following:

$$c_1 = \left(\frac{\Delta x}{6\Delta t^2}\left(\lambda' - 2 + \frac{1}{\lambda'}\right) + \frac{G\Delta x}{12\Delta t}\left(\lambda' - \frac{1}{\lambda'}\right)\right)(A+4)$$

$$c_2 = \frac{gh}{\Delta x}(2-A)\left(\alpha_1\lambda' + \alpha_2 + \alpha_3\frac{1}{\lambda'}\right)$$

$$c_3 = (G-\tau)\frac{h}{2}B$$

$$m_1 = \frac{g}{4}B(\lambda' + 1)$$

$$m_2 = \left[\frac{\Delta x}{\Delta t}(\lambda' - 1) + \frac{\tau\Delta x}{2}(\lambda' + 1)\right]$$

so that the system becomes

$$\begin{bmatrix} c_1 + c_2 & c_3 \\ m_1 & m_2 \end{bmatrix}\begin{bmatrix} \zeta_0 \\ u_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \qquad \textbf{(C.37)}$$

When the determinant is set equal to zero, a cubic polynomial in λ' results

$$P_3\lambda'^3 + P_2\lambda'^2 + P_1\lambda' + P_0 = 0 \qquad \textbf{(C.38)}$$

where the third root is an artifact of the wave equation form of the continuity equation and

has zero phase and the coefficients are given as

$$P_0 = -K_3\left[2+\frac{A}{2}\right]+(K_2(G+\tau)-K_1G\tau)\left[1+\frac{A}{4}\right]+\alpha_3 gh\left[1-\frac{A}{2}\right]\left[\tau-\frac{2}{\Delta t}\right] \quad \textbf{(C.39)}$$

$$P_1 = 3K_3\left[2+\frac{A}{2}\right]-(K_2(G+\tau)+K_1G\tau)\left[1+\frac{A}{4}\right]+$$

$$gh\left[1-\frac{A}{2}\right]\left[\frac{2}{\Delta t}(\alpha_3-\alpha_2)+\tau(\alpha_3+\alpha_2)\right]-\frac{gh}{8}B^2(G-\tau) \quad \textbf{(C.40)}$$

$$P_2 = -3K_3\left[2+\frac{A}{2}\right]-(K_2(G+\tau)-K_1G\tau)\left[1+\frac{A}{4}\right]+$$

$$gh\left[1-\frac{A}{2}\right]\left[\frac{2}{\Delta t}(\alpha_2-\alpha_1)+\tau(\alpha_2+\alpha_1)\right]-\frac{gh}{8}B^2(G-\tau) \quad \textbf{(C.41)}$$

$$P_3 = K_3\left[2+\frac{A}{2}\right]+(K_2(G+\tau)+K_1G\tau)\left[1+\frac{A}{4}\right]+\alpha_1 gh\left[1-\frac{A}{2}\right]\left[\tau+\frac{2}{\Delta t}\right] \quad \textbf{(C.42)}$$

and

$$K_1 = \frac{\Delta x^2}{6\Delta t},\ K_2 = \frac{\Delta x^2}{3\Delta t^2},\ K_3 = \frac{\Delta x^2}{3\Delta t^3}.$$

For brevity, only the cubic polynomial in λ' is given, as the expressions for the roots are tedious. Any software package, such as Mathematica®, can be used to find the roots themselves.

C.1.5 Selective lumping finite element

When the first step is substituted into the second step and like terms are gathered, the discrete equations for the selective lumping finite element (SLFE) discretization of the SWE can be written as

$$\zeta_j^{k+1} = \left[\frac{2+e}{3} - \frac{\mu^2 gh}{4}\right]\zeta_j^k + \frac{1-e}{6}(\zeta_{j-1}^k + \zeta_{j+1}^k) + \frac{\mu^2 gh}{8}(\zeta_{j-2}^k + \zeta_{j+2}^k) -$$

$$\frac{\mu h}{2}\left[\frac{1-e}{6} - \frac{\tau\Delta t}{2}\frac{1-f}{6}\right](u_{j+2}^k - u_{j-2}^k) - \frac{\mu h}{2}\left[\frac{2+e}{3} - \frac{\tau\Delta t}{2}\frac{2+f}{3}\right](u_{j+1}^k - u_{j-1}^k) \qquad \textbf{(C.43)}$$

$$u_j^{k+1} = \left[\frac{\mu^2 gh}{8} - \tau\Delta t\frac{1-e}{6}\frac{1-f}{6} + \frac{\tau^2\Delta t^2}{2}\frac{(1-f)^2}{36}\right](u_{j-2}^k + u_{j+2}^k) +$$

$$\left[\frac{1-e}{6} - \tau\Delta t\left(\frac{1-f}{6}\frac{2+e}{3} + \frac{2+f}{3}\frac{1-e}{6} - \tau\Delta t\frac{1-f}{6}\frac{2+f}{3}\right)\right](u_{j-1}^k + u_{j+1}^k) +$$

$$\left[\frac{2+e}{3} - \frac{\mu^2 gh}{4} + 2\tau\Delta t\frac{1-f}{6}\left(\frac{\tau\Delta t}{2}\frac{1-f}{6} - \frac{1-e}{6}\right) + \tau\Delta t\frac{2+f}{3}\left(\frac{\tau\Delta t}{2}\frac{2+f}{3} - \frac{2+e}{3}\right)\right]u_j^k -$$

$$\frac{\mu g}{2}\left[\frac{1-e}{6} - \frac{\tau\Delta t}{2}\frac{1-f}{6}\right](\zeta_{j+2}^k - \zeta_{j-2}^k) - \frac{\mu g}{2}\left[\frac{2+e}{3} - \frac{\tau\Delta t}{2}\frac{2+f}{3}\right](\zeta_{j+1}^k - \zeta_{j-1}^k) \qquad \textbf{(C.44)}$$

where $\mu = \Delta t/\Delta x$ and Equations (C.43) and (C.44) are not solved in practice, but are only given as a closed form herein for the analytical Fourier analyses.

Now substitute a single Fourier component for each variable, given as

$$\zeta_j^{\,k} = \zeta_0 e^{i\omega k\Delta t} e^{i\sigma j\Delta x} \text{ and } u_j^{\,k} = u_0 e^{i\omega k\Delta t} e^{i\sigma j\Delta x}, \qquad \textbf{(C.45)}$$

divide through by the common factor, $e^{i\omega k\Delta t} e^{i\sigma j\Delta x}$, and write as a system to get

$$\begin{bmatrix} \lambda' - E & \frac{\mu hF}{2} \\ \frac{\mu gF}{2} & \lambda' - E + \tau\Delta tJ \end{bmatrix}\begin{bmatrix}\zeta_0 \\ u_0\end{bmatrix} = \begin{bmatrix}0 \\ 0\end{bmatrix} \qquad \textbf{(C.46)}$$

where λ' and μ are as defined in the previous algorithms and the remaining definitions are as follows:

$$A = e^{i\sigma\Delta x} + e^{-i\sigma\Delta x} = 2\cos(\sigma\Delta x) \qquad \textbf{(C.47)}$$

$$B = e^{i\sigma\Delta x} - e^{-i\sigma\Delta x} = 2i\sin(\sigma\Delta x) \quad \textbf{(C.48)}$$

$$C = e^{i2\sigma\Delta x} + e^{-i2\sigma\Delta x} = 2\cos(2\sigma\Delta x) \quad \textbf{(C.49)}$$

$$D = e^{i2\sigma\Delta x} - e^{-i2\sigma\Delta x} = 2i\sin(2\sigma\Delta x) \quad \textbf{(C.50)}$$

$$E = \frac{2+e}{3} + \frac{1-e}{6}A + \frac{\mu^2 gh}{8}(C-2) \quad \textbf{(C.51)}$$

$$F = \frac{2+e}{3}B + \frac{1-e}{6}D - \frac{\tau\Delta t}{2}\left(\frac{2+f}{3}B + \frac{1-f}{6}D\right) \quad \textbf{(C.52)}$$

$$J = \frac{1-e}{6}\left[\frac{(1-f)}{6}(C+2) + \frac{2+f}{3}A\right] + \frac{2+e}{3}\left[\frac{1-f}{6}A + \frac{2+f}{3}\right] -$$

$$\tau\Delta t\left[\left(\frac{1-f}{6}\right)^2\left(1+\frac{C}{2}\right) + \frac{1-f}{6}\frac{2+f}{3}A + \frac{1}{2}\left(\frac{2+f}{3}\right)^2\right] \quad \textbf{(C.53)}$$

Now set the determinant equal to zero and gather like terms to get the polynomial expression for the Fourier propagation factor,

$$\lambda'^2 + [\tau\Delta tJ - 2E]\lambda' + \left[E^2 - \tau\Delta tEJ - \frac{\mu^2 gh}{4}F^2\right] = 0, \quad \textbf{(C.54)}$$

where after simplification, the roots are found to be

$$\lambda' = \frac{1}{2}[2E - \tau\Delta tJ \pm \sqrt{(\tau\Delta tJ)^2 + gh(\mu F)^2}]. \quad \textbf{(C.55)}$$

C.1.6 Low-order finite volume / discontinuous Galerkin FE

The discrete equations for the low-order FVM and DG approximations to the SWE using explicit time stepping are given as

$$\zeta_j^{k+1} = \zeta_j^k - \frac{\Delta t}{2\Delta x}[h(u_{j+1}^k - u_{j-1}^k) - a(\zeta_{j-1}^k - 2\zeta_j^k + \zeta_{j+1}^k)] \quad \textbf{(C.56)}$$

$$u_j^{k+1} = (1 - \tau\Delta t)u_j^k - \frac{\Delta t}{2\Delta x}[g(\zeta_{j+1}^k - \zeta_{j-1}^k) - a(u_{j-1}^k - 2u_j^k + u_{j+1}^k)] \quad \textbf{(C.57)}$$

where $a^2 = gh$ is the wave celerity.

Now substitute a single Fourier component for each variable, given as

$$\zeta_j^{\,k} = \zeta_0 e^{i\omega k\Delta t} e^{i\sigma j\Delta x} \text{ and } u_j^{\,k} = u_0 e^{i\omega k\Delta t} e^{i\sigma j\Delta x}, \quad \textbf{(C.58)}$$

divide through by the common factor, $e^{i\omega k\Delta t} e^{i\sigma j\Delta x}$, and write as a system to get

$$\begin{bmatrix} \lambda' - 1 - \mu a(A-2)/2 & \mu hB/2 \\ \mu gB/2 & \lambda' - 1 - \mu a(A-2)/2 + \tau\Delta t \end{bmatrix} \begin{bmatrix} \zeta_0 \\ u_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \textbf{(C.59)}$$

where λ', μ, A and B are as defined in the previous algorithms.

Now set the determinant equal to zero and gather like terms to get the polynomial expression for the Fourier propagation factor

$$\lambda'^2 + [-2 - \mu a(A-2) + \tau\Delta t]\lambda' +$$
$$\left[\left(1 + \mu a\frac{(A-2)}{2}\right)^2 - \tau\Delta t\left(1 + \mu a\frac{(A-2)}{2}\right) - gh\left(\frac{\mu B}{2}\right)^2\right] = 0 \quad \textbf{(C.60)}$$

where after simplification, the roots are found to be

$$\lambda' = \frac{1}{2}[2 + \mu a(A-2) - \tau\Delta t \pm \sqrt{(\tau\Delta t)^2 + gh(\mu B)^2}]. \quad \textbf{(C.61)}$$

C.2 Dispersion analysis

C.2.1 Theory

Dispersion analysis is similar to Fourier analysis, except that the time domain remains continuous. The starting point of dispersion analysis is to write the harmonic form of each equation by substituting the temporal harmonic components expressed as

$$\zeta = \hat{\zeta} e^{i\omega t} \text{ and } u = \hat{u} e^{i\omega t}, \tag{C.62}$$

where ω is the temporal frequency, into the differential equations and divide through by the common factor, $e^{i\omega t}$. Here $\hat{\zeta}$ is the spatial harmonic of the elevation and $\hat{u}$ is the spatial harmonic of the velocity. This gives

$$i\omega\hat{\zeta} + h\frac{\partial}{\partial x}\hat{u} = 0 \tag{C.63}$$

$$i\omega\hat{u} + \tau\hat{u} + g\frac{\partial}{\partial x}\hat{\zeta} = 0 \tag{C.64}$$

$$(i\omega)^2\hat{\zeta} + Gi\omega\hat{\zeta} + (G-\tau)h\frac{\partial}{\partial x}\hat{u} - gh\frac{\partial^2}{\partial x^2}\hat{\zeta} = 0 \tag{C.65}$$

for the continuity, momentum and generalized wave continuity equations, respectively.

For the discrete equations, the dispersion characteristics can then be determined by substituting the discrete harmonic spatial solutions, expressed as

$$\hat{\zeta} = \zeta_0 e^{ij\sigma\Delta x} \text{ and } \hat{u} = u_0 e^{ij\sigma\Delta x} \tag{C.66}$$

into the spatial discretizations of Equations (C.63) through (C.65). As in Fourier analysis $i = \sqrt{-1}$, j is the spatial node index, σ is the spatial frequency of the solution, Δx is the discrete spacing of nodes, ζ_0 is the magnitude of the elevation solution, and u_0 is the magnitude of the velocity solution.

The solution of the resulting system of equations for the magnitudes u_0 and ζ_0, which proceeds as in Fourier analysis, determines the relationship between the temporal frequency, ω, and the spatial frequency, σ. The magnitude of ω is then plotted against σ to develop the dispersion curve. As in the Fourier derivations presented above, the full derivation will be given only for the SFD algorithm to show the process and then subsequent derivations will be more abbreviated.

C.2.2 Staggered finite difference

The staggered finite difference discretization of the harmonic equations presented in §C.2.1 is given as

$$i\omega\hat{\zeta}_j + \frac{h}{\Delta x}(\hat{u}_{j+1/2} - \hat{u}_{j-1/2}) = 0 \qquad \textbf{(C.67)}$$

$$(i\omega + \tau)\hat{u}_{j+1/2} + \frac{g}{\Delta x}(\hat{\zeta}_{j+1} - \hat{\zeta}_j) = 0\,. \qquad \textbf{(C.68)}$$

Now substitute the definitions given in Equation (C.66) for the spatial harmonics to get

$$i\omega\zeta_0 e^{ij\sigma\Delta x} + \frac{h}{\Delta x}u_0(e^{i(j+1/2)\sigma\Delta x} - e^{i(j-1/2)\sigma\Delta x}) = 0 \qquad \textbf{(C.69)}$$

$$(i\omega + \tau)u_0 e^{i(j+1/2)\sigma\Delta x} + \frac{g}{\Delta x}\zeta_0(e^{i(j+1)\sigma\Delta x} - e^{ij\sigma\Delta x}) = 0 \qquad \textbf{(C.70)}$$

and divide through by the common factor, $e^{ij\sigma\Delta x}$, to get

$$i\omega\zeta_0 + \frac{h}{\Delta x}u_0(e^{i\sigma\Delta x/2} - e^{-i\sigma\Delta x/2}) = 0 \qquad \textbf{(C.71)}$$

$$(i\omega + \tau)u_0 e^{i\sigma\Delta x/2} + \frac{g}{\Delta x}\zeta_0(e^{i\sigma\Delta x} - 1) = 0\,. \qquad \textbf{(C.72)}$$

Now write as a system of equations in ζ_0 and u_0

$$\begin{bmatrix} i\omega & \frac{h}{\Delta x}(e^{i\sigma\Delta x/2} - e^{-i\sigma\Delta x/2}) \\ \frac{g}{\Delta x}(e^{i\sigma\Delta x} - 1) & (i\omega + \tau)e^{i\sigma\Delta x/2} \end{bmatrix} \begin{bmatrix} \zeta_0 \\ u_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{C.73}$$

and set the determinant equal to zero to get

$$i\omega(i\omega + \tau)e^{i\sigma\Delta x/2} - \frac{gh}{\Delta x^2}(e^{i\sigma\Delta x/2} - e^{-i\sigma\Delta x/2})(e^{i\sigma\Delta x} - 1) = 0, \tag{C.74}$$

which, after gathering like powers of ω, gives the quadratic polynomial in ω to be

$$[-e^{i\sigma\Delta x/2}]\omega^2 + [i\tau e^{i\sigma\Delta x/2}]\omega - \frac{gh}{\Delta x^2}(-2e^{i\sigma\Delta x/2} + e^{-i\sigma\Delta x/2} + e^{i\sigma 3\Delta x/2}) = 0. \tag{C.75}$$

This can be further simplified by dividing through by the leading coefficient to get

$$\omega^2 - i\tau\omega + \frac{gh}{\Delta x^2}(-2 + e^{i\sigma\Delta x} + e^{-i\sigma\Delta x}) = 0, \tag{C.76}$$

which has as its roots

$$\omega' = \frac{i}{2}\left[\tau \pm \sqrt{\tau^2 + 4\frac{gh}{\Delta x^2}(A - 2)}\right] \tag{C.77}$$

where, as in Fourier analysis, $A = e^{i\sigma\Delta x} + e^{-i\sigma\Delta x} = 2\cos\sigma\Delta x$, and the prime notation has been used to differentiate from the continuum dispersion relationship.

C.2.3 Primitive finite element

The primitive finite element discretization of the harmonic equations presented in §C.2.1 is given as

$$i\omega\frac{\Delta x}{6}(\hat{\zeta}_{j-1} + 4\hat{\zeta}_j + \hat{\zeta}_{j+1}) + \frac{h}{2}(\hat{u}_{j+1} - \hat{u}_{j-1}) = 0 \tag{C.78}$$

$$(i\omega+\tau)\frac{\Delta x}{6}(\hat{u}_{j-1}+4\hat{u}_j+\hat{u}_{j+1})+\frac{g}{2}(\hat{\zeta}_{j+1}-\hat{\zeta}_{j-1}) \;=\; 0\,. \qquad \textbf{(C.79)}$$

Now substitute the definitions given in Equation (C.66) for the spatial harmonics, divide through by the common factor, $e^{ij\sigma\Delta x}$, and write as a system of equations to get

$$\begin{bmatrix} i\omega\frac{\Delta x}{6}(3A+2B) & \frac{hB}{2} \\ \frac{gB}{2} & (i\omega+\tau)\frac{\Delta x}{6}(3A+2B) \end{bmatrix}\begin{bmatrix}\zeta_0 \\ u_0\end{bmatrix} \;=\; \begin{bmatrix}0\\0\end{bmatrix} \qquad \textbf{(C.80)}$$

where A is as defined above for the SFD algorithm and $B \;=\; \mathrm{e}^{i\sigma\Delta x}-\mathrm{e}^{-i\sigma\Delta x} \;=\; 2i\sin\sigma\Delta x$ as in the above Fourier analysis derivations.

Now set the determinant equal to zero and gather like powers of ω to get

$$\left[-\left(\frac{\Delta x}{6}(3A+2B)\right)^2\right]\omega^2+\left[i\tau\left(\frac{\Delta x}{6}(3A+2B)\right)^2\right]\omega-\frac{ghB^2}{4} \;=\; 0\,, \qquad \textbf{(C.81)}$$

which can be simplified by dividing through by the coefficient of the leading term

$$\omega^2-i\tau\omega+9gh\left(\frac{B}{\Delta x(3A+2B)}\right)^2 \;=\; 0\,. \qquad \textbf{(C.82)}$$

The roots of this quadratic polynomial in ω are found to be

$$\omega' \;=\; \frac{i}{2}\left[\tau\pm\sqrt{\tau^2+36gh\left(\frac{B}{\Delta x(3A+2B)}\right)^2}\right] \qquad \textbf{(C.83)}$$

where again the prime notation has been used to differentiate from the continuum dispersion relationship.

C.2.4 ADCIRC model (generalized wave continuity)

The ADCIRC model discretization of the harmonic equations presented in §C.2.1

is given as

$$((i\omega)^2 + Gi\omega)\frac{1}{6}(\hat{\zeta}_{j-1} + 4\hat{\zeta}_j + \hat{\zeta}_{j+1}) + (G-\tau)\frac{h}{2\Delta x}(\hat{u}_{j+1} - \hat{u}_{j-1}) -$$

$$\frac{gh}{\Delta x^2}(-\hat{\zeta}_{j-1} + 2\hat{\zeta}_j - \hat{\zeta}_{j+1}) = 0 \qquad \textbf{(C.84)}$$

$$(i\omega + \tau)\hat{u}_j + \frac{g}{2\Delta x}(\hat{\zeta}_{j+1} - \hat{\zeta}_{j-1}) = 0 \qquad \textbf{(C.85)}$$

Now substitute the definitions given in Equation (C.66) for the spatial harmonics, divide through by the common factor, $e^{ij\sigma\Delta x}$, and write as a system of equations to get

$$\begin{bmatrix} ((i\omega)^2 + Gi\omega)\frac{1}{6}(A+4) - \frac{gh}{\Delta x^2}(2-A) & (G-\tau)\frac{hB}{2\Delta x} \\ \frac{gB}{2\Delta x} & (i\omega+\tau) \end{bmatrix} \begin{bmatrix} \zeta_0 \\ u_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \textbf{(C.86)}$$

where A and B are as defined in the previous derivations. Set the determinant equal to zero and gather like powers of ω to arrive at

$$\left[\frac{-i}{6}(4+A)\right]\omega'^3 + \left[\frac{-i}{6}(G+\tau)(4+A)\right]\omega'^2 + \left[\frac{i}{6}G\tau(4+A) + \frac{ihg}{\Delta x^2}(2-A)\right]\omega' +$$

$$\left[\frac{hg\tau}{\Delta x^2}(2-A) + \frac{hg}{4\Delta x^2}(\tau - G)B^2\right] = 0 \qquad \textbf{(C.87)}$$

where the cubic polynomial is a result of the second time derivative, and the prime notation has been used to differentiate from the continuum dispersion relationship. As the roots of a cubic polynomial are rather tedious to present, only the polynomial itself is presented herein. Any mathematical software package, such as Mathematica®, could be used to find the roots themselves.

C.2.5 Selective lumping finite element

No analytical relationship can be determined for this algorithm, due to the two-step temporal discretization scheme. It is impossible to write the equations independently of time, as is required for dispersion analysis, since the temporal discretization is an integral part of the algorithm.

C.2.6 Low-order finite volume / discontinuous Galerkin FE

The low-order finite volume and discontinuous Galerkin finite element discretizations of the harmonic equations presented in §C.2.1 are given as

$$i\omega\hat{\zeta}_j + \frac{1}{2\Delta x}[h(\hat{u}_{j+1} - \hat{u}_{j-1}) - a(\hat{\zeta}_{j-1} - 2\hat{\zeta}_j + \hat{\zeta}_{j+1})] = 0 \qquad \textbf{(C.88)}$$

$$(i\omega + \tau)\hat{u}_j + \frac{1}{2\Delta x}[g(\hat{\zeta}_{j+1} - \hat{\zeta}_{j-1}) - a(\hat{u}_{j-1} - 2\hat{u}_j + \hat{u}_{j+1})] = 0 \qquad \textbf{(C.89)}$$

Now substitute the definitions given in Equation (C.66) for the spatial harmonics, divide through by the common factor, $e^{ij\sigma\Delta x}$, and write as a system of equations to get

$$\begin{bmatrix} i\omega - \frac{a}{2\Delta x}(A-2) & \frac{hB}{2\Delta x} \\ \frac{gB}{2\Delta x} & i\omega + \tau - \frac{a}{2\Delta x}(A-2) \end{bmatrix} \begin{bmatrix} \zeta_0 \\ u_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \textbf{(C.90)}$$

where A and B are as defined in the previous derivations.

Now set the determinant equal to zero and gather like powers of ω to get

$$\omega^2 + \left[-i\tau + \frac{ia}{\Delta x}(A-2)\right]\omega + \left[gh\left(\frac{B}{2\Delta x}\right)^2 + \frac{a}{2\Delta x}(A-2)\left(\tau - \frac{a}{2\Delta x}(A-2)\right)\right] = 0 \qquad \textbf{(C.91)}$$

where the roots of this quadratic polynomial in ω are found to be, after simplification,

$$\omega' = \frac{i}{2}\left[\tau - \frac{a}{\Delta x}(A-2) \pm \sqrt{\tau^2 + gh\left(\frac{B}{\Delta x}\right)^2}\right]. \qquad \textbf{(C.92)}$$

Again the prime notation has been used to differentiate from the continuum dispersion relationship.

APPENDIX D

Spatial and Temporal Grid Convergence Results for 1D SWE Solution Algorithms

D.1 Spatial grid convergence results

The L_2 and L_∞ error norms for the spatial errors, as described in Chapter 3, are summarized for each of the study algorithms herein. A fine grid solution of 1281 nodes (Δx = 39.0625m) from the ADCIRC model is used as the "true" solution. The slope of a best-fit line through this data in log-log space gives the order of the approximation, or the convergence rate. Also, the slopes calculated between two successive grid resolutions are used to compute average and peak convergence rates. The best-fit and average convergence rates are computed for the linear portions of the data; the flat portion of the curves is not included. These results are presented graphically in Chapter 3 and the raw data is provided here in Tables D.1 through D.10.

Table D.1 Spatial grid convergence results for 1D SFD algorithm.

	L_∞ error norms		L_2 error norms	
Δx (m)	velocity	elevation	velocity	elevation
5000	0.01111999	0.03948002	0.00529831	0.03293353
2500	0.00452118	0.02114269	0.00242530	0.01750980
1250	0.00208092	0.01056154	0.00130898	0.00907054
625	0.00111577	0.00531280	0.00069661	0.00462047
312.5	0.00064611	0.00265998	0.00036144	0.00233179
156.25	0.00035477	0.00133057	0.00018492	0.00117097
78.125	0.00021681	0.00066552	0.00009430	0.00058639
39.0625	0.00015993	0.00033168	0.00004834	0.00029306
best-fit rate	0.931	0.990	0.956	0.977
average rate	0.947	0.985	0.968	0.973
peak rate	1.298	1.005	1.127	1.001

Table D.2 Spatial grid convergence results for 1D PLF algorithm.

	L_∞ error norms		L_2 error norms	
Δx (m)	velocity	elevation	velocity	elevation
5000	0.00043724	0.00085701	0.00025002	0.00049621
2500	0.00024993	0.00025060	0.00009464	0.00013408
1250	0.00013701	0.00009190	0.00003580	0.00003944
625	0.00010862	0.00004497	0.00001742	0.00001327
312.5	0.00010260	0.00002309	9.954066e-6	5.413497e-6
156.25	0.00010445	0.00001731	6.917866e-6	3.335049e-6
78.125	0.00010361	0.00001818	5.395245e-6	3.070488e-6
39.0625	0.00010357	0.00001973	4.662774e-6	3.064374e-6
best-fit rate	0.837	1.291	1.174	1.637
average rate	0.837	1.304	1.163	1.630
peak rate	0.867	1.774	1.402	1.888

Table D.3 Spatial grid convergence results for 1D ADCIRC model.

	L_∞ error norms		L_2 error norms	
Δx (m)	velocity	elevation	velocity	elevation
5000	0.01282147	0.002063137	0.004459657	0.00151752
2500	0.00623724	0.00049856	0.00159062	0.00039099
1250	0.00295148	0.00017966	0.00055701	0.00011083
625	0.00146198	0.00013991	0.00019680	0.00004777
312.5	0.00072114	0.00007972	0.00006958	0.00002114
156.25	0.00031039	0.00004775	0.00002366	9.93729e-6
78.125	0.00013054	0.00002747	7.99160e-6	4.49712e-6
best-fit rate	1.091	1.518	0.951	1.364
average rate	1.103	1.521	1.038	1.400
peak rate	1.250	1.566	2.049	1.957

Table D.4 Spatial grid convergence results for 1D SLFE algorithm.

	L_∞ error norms		L_2 error norms	
Δx (m)	velocity	elevation	velocity	elevation
5000	0.00414089	0.01641770	0.00163751	0.01411178
2500	0.00237912	0.00891522	0.00101341	0.00775322
1250	0.00124155	0.00447967	0.00054521	0.00406424
625	0.00066050	0.00227085	0.00028214	0.00208352
312.5	0.00037726	0.00114595	0.00014485	0.00105492
156.25	0.00022007	0.00059840	0.00007402	0.00053094
78.125	0.00013125	0.00032282	0.00003761	0.00026674
39.0625	0.00011092	0.00017582	0.00001901	0.00013412
best-fit rate	0.840	0.946	0.959	0.966
average rate	0.830	0.935	0.956	0.960
peak rate	0.938	0.993	0.984	0.993

Table D.5 Spatial grid convergence results for 1D low-order FVM/DG.

	L_∞ error norms		L_2 error norms	
Δx (m)	velocity	elevation	velocity	elevation
5000	0.03702352	0.16967875	0.02086256	0.13968249
2500	0.02070707	0.08454615	0.01138230	0.06979790
1250	0.01085377	0.04283867	0.00580967	0.03516845
625	0.00556164	0.02188525	0.00299509	0.01775876
312.5	0.00281441	0.01109734	0.00157794	0.00893168
156.25	0.00141434	0.00557090	0.00082827	0.00447561
78.125	0.00070876	0.00278744	0.00043048	0.00223841
39.0625	0.00035477	0.00139351	0.00022394	0.00111910
best-fit rate	0.965	0.987	0.938	0.994
average rate	0.958	0.990	0.935	0.995
peak rate	0.998	1.005	0.970	1.001

Table D.6 Spatial grid convergence results for 1D minmod limited FVM.

	L_∞ error norms		L_2 error norms	
Δx (m)	velocity	elevation	velocity	elevation
5000.	0.06986880	0.01710693	0.02499239	0.01351432
2500.	0.03438994	0.01005677	0.01031451	0.00810453
1250.	0.01722827	0.00523638	0.00449823	0.00436109
625.	0.00861693	0.00262424	0.00205627	0.00224514
312.5	0.00421749	0.00133579	0.00097299	0.00113861
156.25	0.00213754	0.00067208	0.00047445	0.00057525
78.125	0.00100430	0.00033235	0.00023417	0.00028929
39.0625	0.00050427	0.00016580	0.00011724	0.00014542
best-fit rate	1.016	0.968	1.098	0.947
average rate	1.016	0.956	1.105	0.934
peak rate	1.090	1.016	1.277	0.992

Table D.7 Spatial grid convergence results for 1D superbee limited FVM.

	L_∞ error norms		L_2 error norms	
Δx (m)	velocity	elevation	velocity	elevation
5000	0.05940099	0.03604274	0.01960381	0.02710818
2500	0.02848705	0.01654811	0.00692049	0.01351983
1250	0.01430583	0.00781652	0.00250782	0.00670120
625	0.00712807	0.00376295	0.00089999	0.00333100
312.5	0.00333537	0.00186575	0.00030859	0.00166541
156.25	0.00170634	0.00094087	0.00011611	0.00083530
78.125	0.00078590	0.00046922	0.00004206	0.00041834
39.0625	0.00039486	0.00023446	0.00001650	0.00020975
best-fit rate	1.034	1.032	1.466	1.002
average rate	1.033	1.038	1.459	1.002
peak rate	1.118	1.123	1.544	1.013

Table D.8 Spatial grid convergence results for 1D vanLeer limited FVM.

	L_∞ error norms		L_2 error norms	
Δx (m)	velocity	elevation	velocity	elevation
5000	0.06264975	0.14341112	0.03740583	0.10280013
2500	0.04004857	0.08956832	0.02140672	0.06162252
1250	0.02266951	0.05103670	0.01176175	0.03442155
625	0.01203567	0.02743747	0.00626120	0.01826778
312.5	0.00621730	0.01427110	0.00327795	0.00941353
156.25	0.00319331	0.00727676	0.00169178	0.00477600
78.125	0.00161718	0.00367441	0.00086311	0.00240475
39.0625	0.00081366	0.00184683	0.00043817	0.00120671
best-fit rate	0.943	0.940	0.938	0.951
average rate	0.937	0.933	0.935	0.946
peak rate	0.991	0.992	0.978	0.995

Table D.9 Spatial grid convergence results for 1D unlimited DG.

	L_∞ error norms		L_2 error norms	
Δx (m)	velocity	elevation	velocity	elevation
5000	0.00165234	0.00094039	0.00111539	0.00051554
2500	0.00051592	0.00029017	0.00029463	0.00012622
1250	0.00016760	0.00010764	0.00007901	0.00003558
625	0.00012173	0.00004463	0.00002593	0.00001213
312.5	0.00010833	0.00002323	0.00001170	4.88555e-6
156.25	0.00010497	0.00001543	7.31394e-6	2.76321e-6
78.125	0.00010411	0.00002024	5.96062e-6	3.26047e-6
39.0625	0.00010389	0.00003423	5.40672e-6	4.10014e-6
best-fit rate	1.651	1.338	1.666	1.682
average rate	1.651	1.335	1.644	1.680
peak rate	1.679	1.696	1.921	1.696

Table D.10 Spatial grid convergence results for 1D minmod limited DG.

	L_∞ error norms		L_2 error norms	
Δx (m)	velocity	elevation	velocity	elevation
5000	0.04466828	0.08901955	0.01577199	0.06572041
2500	0.02142211	0.04364349	0.00720095	0.03164476
1250	0.01060520	0.02158032	0.00356593	0.01554677
625	0.00524778	0.01073390	0.00177972	0.00771668
312.5	0.00259691	0.00534826	0.00088954	0.00384458
156.25	0.00123695	0.00267072	0.00044475	0.00191870
78.125	0.00056800	0.00133420	0.00022307	0.00095790
39.0625	0.00023221	0.00066681	0.00011224	0.00047807
best-fit rate	1.067	1.008	1.012	1.012
average rate	1.084	1.009	1.019	1.015
peak rate	1.290	1.028	1.131	1.054

D.2 Temporal interval halving results

The L_2 and L_∞ error norms for the temporal errors, as described in Chapter 3, are summarized for each of the study algorithms herein. Interval halving, wherein successive solutions have a time step equal to one-half of the previous time step, was used to generate the convergence results. Except for the SLFE algorithm, a base time step of Δt_b = 1.6 seconds was used for all algorithms, such that the errors are computed for two successive solutions with time steps of $\Delta t_b/m$ and $\Delta t_b/2m$. A base time step of Δt_b = 2.8125 seconds was used for the SLFE algorithm, since larger time steps are required for consistent results. Additionally, some of the discontinuous solutions are unstable at these large time steps, as will be noted in the tables where the lower m values will be missing. These results are presented graphically in Chapter 3 and the raw data is provided here in Tables D.11 through D.20. Note that the ratio of successive error norms is equal to 2^p, where p is the order of the approximation when this ratio converges to a constant value. Unless otherwise noted, the 2^p entries for the last m value within each table are used to compute the convergence rates, p.

Table D.11 Temporal convergence results for 1D SFD algorithm.

	L_∞ error norms				L_2 error norms			
m	velocity	2^p	elevation	2^p	velocity	2^p	elevation	2^p
1	3.82528e-6	—	3.59725e-6	—	1.49512e-6	—	1.50362e-6	—
2	1.92341e-6	1.99	2.18063e-6	1.65	7.01211e-7	2.13	7.59658e-7	1.98
4	9.79565e-7	1.96	1.05851e-6	2.06	3.42537e-7	2.05	3.42985e-7	2.21
8	5.14603e-7	1.90	4.31055e-7	2.46	1.42825e-7	2.40	1.38044e-7	2.48
16	1.18150e-7	4.36	1.19009e-7	3.62	3.86751e-8	3.69	3.92028e-8	3.52
32	3.20321e-8	3.69	3.14840e-8	3.78	9.67587e-9	4.00	9.93065e-9	3.95
p		**1.88**		**1.92**		**2.00**		**1.98**

Table D.12 Temporal convergence results for 1D PLF algorithm.

	L_∞ error norms				L_2 error norms			
m	velocity	2^p	elevation	2^p	velocity	2^p	elevation	2^p
1	0.00002010	—	0.00001754	—	7.08424e-6	—	0.00001167	—
2	0.00001319	1.52	0.00001184	1.48	3.53910e-6	2.00	5.82872e-6	2.00
4	4.58450e-6	2.88	4.62000e-6	2.56	1.77092e-6	2.00	2.91569e-6	2.00
8	2.10150e-6	2.18	2.56794e-6	1.80	8.92226e-7	1.98	1.46657e-6	1.99
16	9.85417e-7	2.13	1.31648e-6	1.95	4.35284e-7	2.05	7.25394e-7	2.02
32	4.22700e-7	2.33	5.47254e-7	2.41	2.11098e-7	2.06	3.58152e-7	2.03
p		**1.22**		**1.27**		**1.04**		**1.02**

Table D.13 Temporal convergence results for 1D ADCIRC model.

	L_∞ error norms				L_2 error norms			
m	velocity	2^p	elevation	2^p	velocity	2^p	elevation	2^p
1	3.79258e-6	—	3.82580e-6	—	8.58070e-7	—	7.38999e-7	—
2	9.69215e-7	3.91	9.28063e-7	4.12	2.27812e-7	3.77	1.93958e-7	3.81
*4**	*2.43831e-7*	*3.97*	*2.40991e-7*	*3.85*	*5.60527e-8*	*4.06*	*4.98683e-8*	*3.89*
8	1.33203e-7	1.83	6.17263e-8	3.90	5.28166e-8	1.06	2.95586e-8	1.69
16	8.43224e-8	1.58	3.50803e-8	1.76	4.84008e-8	1.09	2.55297e-8	1.16
32	7.47700e-7	0.11	3.72498e-7	0.09	5.15810e-7	0.09	2.70534e-7	0.09
p		**1.99**		**1.95**		**2.02**		**1.96**

* Converged accuracy taken from italicized row.

Table D.14 Temporal convergence results for 1D SLFE algorithm.

	L_∞ error norms				L_2 error norms			
m	velocity	2^p	elevation	2^p	velocity	2^p	elevation	2^p
1/2	0.00008645	—	0.00013421	—	0.00003771	—	0.00011844	—
1	0.00014551	0.59	0.00005706	2.35	0.00007359	0.51	0.00004773	2.48
2	0.00025197	0.58	0.00007806	0.73	0.00014521	0.51	0.00002922	1.63
4	0.00045528	0.55	0.00014138	0.55	0.00028829	0.50	0.00007078	0.41
8	0.00093557	0.49	0.00021437	0.66	0.00057363	0.50	0.00013673	0.52
16	0.00185184	0.51	0.00031224	0.69	0.00113890	0.50	0.00024205	0.56
p		**-0.99**		**-0.54**		**-0.99**		**-0.82**

Table D.15 Temporal convergence results for 1D low-order FVM/DG.

	L_∞ error norms				L_2 error norms			
m	velocity	2^p	elevation	2^p	velocity	2^p	elevation	2^p
1	0.00004726	—	0.00002701	—	0.00002857	—	0.00001908	—
2	0.00002319	2.04	0.00001351	2.00	0.00001423	2.01	9.36754e-6	2.04
4	0.00001151	2.02	6.75278e-6	2.00	7.10251e-6	2.00	4.64376e-6	2.02
8	5.73144e-6	2.01	3.37641e-6	2.00	3.54813e-6	2.00	2.31229e-6	2.01
16	2.86068e-6	2.00	1.68821e-6	2.00	1.77329e-6	2.00	1.15380e-6	2.00
32	1.42911e-6	2.00	8.44108e-7	2.00	8.86456e-7	2.00	5.76317e-7	2.00
64	7.14247e-7	2.00	4.22054e-7	2.00	4.43181e-7	2.00	2.88014e-7	2.00
p		**1.00**		**1.00**		**1.00**		**1.00**

Table D.16 Temporal convergence results for 1D minmod limited FVM.

	L_∞ error norms				L_2 error norms			
m	velocity	2^p	elevation	2^p	velocity	2^p	elevation	2^p
1	unstable	—	unstable	—	unstable	—	unstable	—
2	0.00722223	—	0.00695847	—	0.00213639	—	0.00200383	—
4	0.00025100	28.77	0.00023253	29.92	0.00004329	49.35	0.00004271	46.91
8	0.00005300	4.74	0.00005137	4.53	9.23198e-6	4.69	9.03529e-6	4.73
16	0.00001951	2.72	0.00001820	2.82	3.75781e-6	2.46	3.59227e-6	2.52
32	8.57388e-6	2.28	7.90889e-6	2.30	1.73020e-6	2.17	1.63722e-6	2.19
64	4.03972e-6	2.12	3.70330e-6	2.14	8.34583e-7	2.07	7.85413e-7	2.08
p		**1.09**		**1.09**		**1.05**		**1.06**

Table D.17 Temporal convergence results for 1D superbee limited FVM.

	L_∞ error norms				L_2 error norms			
m	velocity	2^p	elevation	2^p	velocity	2^p	elevation	2^p
1	unstable	—	unstable	—	unstable	—	unstable	—
2	unstable	—	unstable	—	unstable	—	unstable	—
4	unstable	—	unstable	—	unstable	—	unstable	—
8	unstable	—	unstable	—	unstable	—	unstable	—
16	0.00004485	—	0.00004325	—	9.47990e-6	—	9.43380e-6	—
32	0.00001330	3.37	0.00001262	3.43	2.64701e-6	3.58	2.58840e-6	3.64
64	5.57205e-6	2.39	5.22455e-6	2.41	1.09685e-6	2.41	1.06002e-6	2.44
p		**1.26**		**1.27**		**1.27**		**1.29**

Table D.18 Temporal convergence results for 1D vanLeer limited FVM.

	L_∞ error norms				L_2 error norms			
m	velocity	2^p	elevation	2^p	velocity	2^p	elevation	2^p
1	0.00005822	—	0.00005933	—	0.00003483	—	0.00002721	—
2	0.00002351	2.48	0.00001866	3.18	0.00001454	2.40	9.10707e-6	2.99
4	0.00001163	2.02	6.34845e-6	2.94	7.14643e-6	2.03	4.32367e-6	2.11
8	5.80605e-6	2.00	3.17668e-6	2.00	3.57060e-6	2.00	2.15496e-6	2.01
16	2.90075e-6	2.00	1.58823e-6	2.00	1.78461e-6	2.00	1.07549e-6	2.00
32	1.44982e-6	2.00	7.94088e-7	2.00	8.92132e-7	2.00	5.37250e-7	2.00
64	7.24777e-7	2.00	3.97036e-7	2.00	4.46023e-7	2.00	2.68502e-7	2.00
p		**1.00**		**1.00**		**1.00**		**1.00**

Table D.19 Temporal convergence results for 1D unlimited DG.

	L_∞ error norms				L_2 error norms			
m	velocity	2^p	elevation	2^p	velocity	2^p	elevation	2^p
1 - 8	unstable	—	unstable	—	unstable	—	unstable	—
16	0.02395420	—	0.02418774	—	0.00357125	—	0.00360566	—
32	0.00008741	274.0	0.00008747	276.5	9.9657e-6	358.4	0.00001004	359.0
64	0.00001681	5.20	0.00001659	5.27	1.9097e-6	5.22	1.9036e-6	5.28
128	6.2593e-6	2.69	6.12607e-6	2.71	7.4590e-7	2.56	7.3735e-7	2.58
256	2.7789e-6	2.25	2.7090e-6	2.26	3.4084e-7	2.19	3.3550e-7	2.20
512	1.3168e-6	2.11	1.2811e-6	2.11	1.6393e-7	2.08	1.6103e-7	2.08
p		**1.08**		**1.08**		**1.06**		**1.06**

Table D.20 Temporal convergence results for 1D minmod limited DG.

	L_∞ error norms				L_2 error norms			
m	velocity	2^p	elevation	2^p	velocity	2^p	elevation	2^p
1	10.0791892	—	10.6250467	—	4.74312272	—	4.18033528	—
2	0.00611484	—	0.00625862	—	0.00181409	—	0.00197993	—
4	0.00021283	28.7	0.00024122	25.9	0.00003931	46.1	0.00004505	44.0
8	0.00004716	4.51	0.00004686	5.15	8.5968e-6	4.57	9.8283e-6	4.58
16	0.00001709	2.76	0.00001675	2.80	3.4319e-6	2.50	4.2386e-6	2.32
32	7.4825e-6	2.28	7.3330e-6	2.28	1.5564e-6	2.21	2.0044e-6	2.11
64	3.5537e-6	2.11	3.4790e-6	2.11	7.4661e-7	2.08	9.8732e-7	2.03
128	1.7853e-6	1.99 *	1.7427e-6	2.00	3.7866e-7	1.97 *	4.9823e-7	1.98
256	1.0432e-6	1.71	9.0457e-7	1.93	2.4019e-7	1.58	2.4344e-7	2.05
p		**0.99**		**0.95**		**0.99**		**1.03**
The converged ratio for the velocity errors are taken from the m = 128 data point.								

APPENDIX E

Spatial and Temporal Convergence Results for 1D Coupled Discontinuous and Continuous SWE Solution Algorithms

E.1 Spatial grid convergence results

The L_2 and L_∞ error norms for the spatial errors, as described in Chapter 4, are summarized for each of the coupled algorithms herein. A fine grid solution of 1281 nodes (Δx = 39.0625m) from the ADCIRC model is used as the "true" solution. A time step of 0.01 seconds was used for all simulations, including the "true" solution. The accuracy, or order of the approximation, for each model is approximated by three different computations: 1) "best-fit" is the slope of the best fit line through the linear portion of the log-log plots, 2) "average" is the average of the slopes between successive data points in log-log space, and 3) "peak" is the maximum of the slopes between successive data points in log-log space. The log-log plots themselves are presented in Chapter 4 and the raw data and approximated orders are provided here in Tables E.1 through E.9.

Table E.1 Spatial convergence results for 1D equation coupled DG/CG model with piecewise constant approximations.

	L_∞ error norms		L_2 error norms	
Δx (m)	velocity	elevation	velocity	elevation
5000	0.03507000	0.00781361	0.02075245	0.00347168
2500	0.01919184	0.00263946	0.01126178	0.00166990
1250	0.00993716	0.00153651	0.00588286	0.00106423
625	0.00501104	0.00106836	0.00301080	0.00066230
312.5	0.00249495	0.00077764	0.00152466	0.00039960
156.25	0.00135526	0.00055177	0.00076848	0.00023514
78.125	0.00076807	0.00037901	0.00038732	0.00013481
39.0625	0.00044782	0.00025594	0.00019570	0.00007627
best-fit	0.915	0.636	0.967	0.762
average	0.899	0.705	0.961	0.787
peak	1.006	1.566	0.988	1.056

Table E.2 Spatial convergence results for 1D equation coupled DG/CG model with unlimited piecewise linear approximations.

	L_∞ error norms		L_2 error norms	
Δx (m)	velocity	elevation	velocity	elevation
5000	0.00196626	0.00453322	0.00126043	0.00253494
2500	0.00092744	0.00120015	0.00049588	0.00072467
1250	0.00046570	0.00045641	0.00021250	0.00024974
625	0.00022662	0.00023862	0.00010042	0.00010471
312.5	0.00013831	0.00013820	0.00005056	0.00004947
156.25	0.00010424	0.00008822	0.00002537	0.00002501
78.125	0.00010361	0.00005440	0.00001452	0.00001384
39.0625	0.21180233	0.17422598	0.05236029	0.06302724
best-fit	0.969	1.014	1.070	1.236
average	0.957	1.063	1.073	1.253
peak	1.084	1.917	1.346	1.807

Table E.3 Spatial convergence results for 1D equation coupled DG/CG model with minmod limited piecewise linear approximations.

	L_∞ error norms		L_2 error norms	
Δx (m)	velocity	elevation	velocity	elevation
5000	0.01005670	0.00808653	0.00423833	0.00344977
2500	0.00449376	0.00316799	0.00159232	0.00102661
1250	0.00193015	0.00130298	0.00054665	0.00033274
625	0.00082952	0.00055838	0.00017610	0.00011593
312.5	0.00033744	0.00024699	0.00005860	0.00004575
156.25	0.00012015	0.00011197	0.00002183	0.00001997
78.125	0.00004181	0.00005198	9.7091e-6	9.7237e-6
39.0625	0.00008274	0.00002791	6.2335e-6	5.4820e-6
best-fit	1.311	1.175	1.497	1.336
average	1.318	1.168	1.462	1.328
peak	1.523	1.352	1.634	1.749

Table E.4 Spatial convergence results for 1D coupled DG –> ADCIRC model with piecewise constant approximations.

	L_∞ error norms		L_2 error norms	
Δx (m)	velocity	elevation	velocity	elevation
5000	0.08206212	0.05452019	0.05290144	0.03634489
2500	0.04205397	0.02741303	0.02669914	0.01769818
1250	0.02164133	0.01419396	0.01361186	0.00893717
625	0.01114768	0.00726121	0.00691758	0.00450817
312.5	0.00573852	0.00365672	0.00349675	0.00226765
156.25	0.00293296	0.00183685	0.00176131	0.00113983
78.125	0.00148093	0.00092270	0.00088645	0.00057294
39.0625	0.00074224	0.00046275	0.00044583	0.00028845
best-fit	0.967	0.982	0.984	0.994
average	0.970	0.983	0.984	0.997
peak	0.997	0.996	0.992	1.038

Table E.5 Spatial convergence results for 1D coupled DG –> ADCIRC model with unlimited piecewise linear approximations.

	L_∞ error norms		L_2 error norms	
Δx (m)	velocity	elevation	velocity	elevation
5000	0.01212657	0.00318933	0.00419702	0.00205673
2500	0.00602124	0.00080762	0.00153532	0.00052103
1250	0.00295208	0.00018389	0.00054927	0.00013348
625	0.00150788	0.00008211	0.00019733	0.00004032
312.5	0.00068417	0.00005244	0.00006677	0.00001514
156.25	0.00029534	0.00003987	0.00002213	6.7107e-6
78.125	0.00012972	0.00001391	7.2655e-6	2.3721e-6
39.0625	0.00001327	0.00001297	2.5507e-6	2.2302e-6
best-fit	1.281	1.215	1.534	1.606
average	1.405	1.307	1.526	1.627
peak	3.289	2.135	1.607	1.981

Table E.6 Spatial convergence results for 1D coupled DG –> ADCIRC model with minmod limited piecewise linear approximations.

	L_∞ error norms		L_2 error norms	
Δx (m)	velocity	elevation	velocity	elevation
5000	0.03547245	0.03623953	0.02081462	0.01321169
2500	0.02052077	0.01726474	0.01238833	0.00512514
1250	0.01123804	0.00842210	0.00683957	0.00219122
625	0.00584387	0.00411988	0.00359850	0.00105387
312.5	0.00295772	0.00203585	0.00184567	0.00052417
156.25	0.00148633	0.00101032	0.00093493	0.00026191
78.125	0.00074591	0.00050259	0.00047144	0.00013110
39.0625	0.00037399	0.00025075	0.00023697	0.00006571
best-fit	0.948	1.023	0.933	1.074
average	0.938	1.025	0.922	1.093
peak	0.996	1.070	0.992	1.366

Table E.7 Spatial convergence results for 1D coupled DG/CG –> ADCIRC model with piecewise constant approximations.

	L_∞ error norms		L_2 error norms	
Δx (m)	velocity	elevation	velocity	elevation
5000	0.08093345	0.03341923	0.06521852	0.01955394
2500	0.03669232	0.01038929	0.02897435	0.00459859
1250	0.01835309	0.00526200	0.01466590	0.00272636
625	0.00886315	0.00247034	0.00714362	0.00140816
312.5	0.00423990	0.00113818	0.00349966	0.00069951
156.25	0.00200646	0.00053660	0.00173263	0.00035005
78.125	0.00106237	0.00033331	0.00086000	0.00017698
39.0625	0.00057229	0.00020988	0.00042774	0.00009015
best-fit	1.026	1.036	1.029	1.044
average	1.021	1.045	1.036	1.109
peak	1.141	1.686	1.171	2.088

Table E.8 Spatial convergence results for 1D coupled DG/CG –> ADCIRC model with unlimited piecewise linear approximations.

	L_∞ error norms		L_2 error norms	
Δx (m)	velocity	elevation	velocity	elevation
5000	0.13225503	0.05906298	0.10760493	0.04242782
2500	1.30246895	3.07591694	0.58595958	1.58533668
1250	0.20795472	0.68271497	0.09815036	0.32691450
625	6.64994609	5.97165529	2.17561222	2.62063902
312.5	9.03342258	8.10267666	2.91560887	2.98131633
156.25	3.95839057	3.25415539	0.99317998	1.24692051
78.125	90.3073024	79.8062710	20.1016227	22.4564965
39.0625	unstable	unstable	unstable	unstable

* Due to the instability of this algorithm, convergence rates are meaningless.

Table E.9 Spatial convergence results for 1D coupled DG/CG –> ADCIRC model with minmod limited piecewise linear approximations.

	L_∞ error norms		L_2 error norms	
Δx (m)	velocity	elevation	velocity	elevation
5000	0.07400487	0.03424017	0.05779450	0.01656934
2500	0.03284014	0.01222124	0.02508870	0.00435227
1250	0.01594067	0.00594885	0.01337371	0.00261822
625	0.00791121	0.00284464	0.00679698	0.00142056
312.5	0.00402146	0.00134173	0.00343982	0.00073024
156.25	0.00201634	0.00063365	0.00173871	0.00036346
78.125	0.00098013	0.00030092	0.00087525	0.00017880
39.0625	0.00046628	0.00014478	0.00043967	0.00008839
best-fit	1.029	1.104	0.992	1.017
average	1.044	1.127	1.005	1.079
peak	1.172	1.486	1.204	1.929

E.2 Temporal interval halving results

The L_2 and L_∞ error norms for the temporal errors, as described in Chapter 4, are summarized for each of the coupled study algorithms herein. Interval halving, wherein successive solutions have a time step equal to one-half of the previous time step, was used to generate the convergence results. A base time step of $\Delta t_b = 1.6$ seconds was used for all algorithms, such that the errors are computed for two successive solutions with time steps of $\Delta t_b/m$ and $\Delta t_b/2m$. Some of the coupled solutions are unstable at these large time steps, as will be noted in the tables where the lower m values will be missing. These results are presented graphically in Chapter 4 and the raw data is provided here in Tables E.10 through E.18. Note that the ratio of successive error norms is equal to 2^p, where p is the order of the approximation when this ratio converges to a constant value. Unless otherwise noted, the 2^p entries for the last m value within each table are used to compute the convergence rates, p.

Table E.10 Temporal convergence results for 1D equation coupled DG/CG model with piecewise constant approximations.

	L_∞ error norms				L_2 error norms			
m	velocity	2^p	elevation	2^p	velocity	2^p	elevation	2^p
1	0.00003346	—	0.00001818	—	0.00001807	—	9.7121e-6	—
2	0.00001633	2.05	8.6179e-6	2.11	9.0039e-6	2.01	4.7360e-6	2.05
4	8.0780e-6	2.02	4.2001e-6	2.05	4.4946e-6	2.00	2.3397e-6	2.02
8	4.0183e-6	2.01	2.0738e-6	2.03	2.2456e-6	2.00	1.1630e-6	2.01
16	2.0041e-6	2.01	1.0305e-6	2.01	1.1224e-6	2.00	5.7980e-7	2.01
32	1.0008e-6	2.00	5.1365e-7	2.01	5.6108e-7	2.00	2.8948e-7	2.00
p		**1.00**		**1.00**		**1.00**		**1.00**

Table E.11 Temporal convergence results for 1D equation coupled DG/CG model with unlimited piecewise linear approximations.

	L_∞ error norms				L_2 error norms			
m	velocity	2^p	elevation	2^p	velocity	2^p	elevation	2^p
1-32	unstable	—	unstable	—	unstable	—	unstable	—
64	0.00380814	—	0.00254330	—	0.00081105	—	0.00081860	—
128	0.00005026	75.77	0.00002974	85.53	0.00001007	80.53	0.00001019	80.32
256	4.8532e-6	10.36	3.0776e-6	9.66	1.0366e-6	9.72	1.0492e-6	9.71
512	1.1276e-6	4.30	8.8326e-7	3.48	2.7695e-7	3.74	2.8009e-7	3.75
1024	3.9019e-7	2.89	3.4566e-7	2.56	1.0797e-7	2.57	1.0897e-7	2.57
2048	1.6609e-7	2.35	1.5413e-7	2.24	4.8522e-8	2.23	4.8895e-8	2.23
p		**1.23**		**1.17**		**1.15**		**1.16**

Table E.12 Temporal convergence results for 1D equation coupled DG/CG model with minmod limited piecewise linear approximations.

	L_∞ error norms				L_2 error norms			
m	velocity	2^p	elevation	2^p	velocity	2^p	elevation	2^p
1	4.43040325	—	4.53973911	—	1.28788171	—	1.34285149	—
2	0.00171690	—	0.001984570	—	0.00065100	—	0.00068380	—
4	0.00005701	30.1	0.00006047	32.8	0.00002028	32.1	0.00001881	36.4
8	0.00001923	2.97	0.00001743	3.47	5.2596e-6	3.86	4.8000e-6	3.92
16	8.6574e-6	2.22	7.7852e-6	2.24	2.0878e-6	2.52	1.8895e-6	2.54
32	4.1130e-6	2.10	3.677-e-6	2.12	9.8298e-7	2.12	8.7346e-7	2.16
64	2.0311e-6	2.03	1.81239e-6	2.03	4.7731e-7	2.06	4.1909e-7	2.08
128	1.0749e-6	1.89	9.5689e-7	1.89	2.4188e-7	1.97	2.0992e-7	2.00
256	6.1045e-7	1.76	5.3868e-7	1.78	1.6011e-7	1.51	1.3984e-7	1.50
p		**0.92**		**0.92**		**0.98**		**1.00**

Table E.13 Temporal convergence results for 1D subdomain coupled DG -> ADCIRC model with piecewise constant approximations.

	L_∞ error norms				L_2 error norms			
m	velocity	2^p	elevation	2^p	velocity	2^p	elevation	2^p
1	0.00003097	—	0.00002001	—	0.00001638	—	0.00001047	—
2	0.00001506	2.06	9.5524e-6	2.10	8.1478e-6	2.01	5.1259e-6	2.04
4	7.4351e-6	2.03	4.6717e-6	2.04	4.0664e-6	2.00	2.5376e-6	2.02
8	3.7000e-6	2.01	2.3092e-6	2.02	2.0343e-6	2.00	1.2629e-6	2.01
16	1.8796e-6	1.97	1.1366e-6	2.03	1.0391e-6	1.96	6.3270e-7	2.00
32	9.6688e-7	1.94	5.5717e-7	2.04	5.3762e-7	1.93	3.1867e-7	1.99
64	8.1586e-7	1.19	3.4592e-7	1.61	4.7984e-7	1.12	2.4416e-7	1.31
128	5.3963e-6	0.15	2.4323e-6	0.14	3.0180e-6	0.16	2.0932e-6	0.12
256	0.00001240	0.44	5.6035e-6	0.43	6.8577e-6	0.44	4.8883e-6	0.43
p *		**0.96**		**1.03**		**0.95**		**0.99**

* Converged rate taken from *m* = 32 ratio before model begins to diverge due to ADCIRC BCs.

Table E.14 Temporal convergence results for 1D subdomain coupled DG -> ADCIRC model with unlimited piecewise linear approximations.

	L_∞ error norms				L_2 error norms			
m	velocity	2^p	elevation	2^p	velocity	2^p	elevation	2^p
1-4	unstable	—	unstable	—	unstable	—	unstable	—
8	8.79937283	—	8.85318433	—	2.20323573	—	2.25141420	—
16	0.00009887	—	0.00008242	—	0.00001813	—	0.00001401	—
32	7.9132e-6	12.5	7.4993e-6	11.0	1.9004e-6	9.54	1.7904e-6	7.83
64	3.9371e-6	2.01	3.3285e-6	2.25	8.1874e-7	2.32	7.0945e-7	2.52
128	6.3671e-6	0.62	2.5735e-6	1.29	3.0389e-6	0.27	2.1121e-6	0.34
256	0.00001237	0.51	5.6522e-6	0.46	6.8663e-6	0.44	4.8865e-6	0.43
p *		**1.01**		**1.17**		**1.21**		**1.34**

* Converged rate taken from *m* = 64 ratio before model begins to diverge due to ADCIRC BCs.

Table E.15 Temporal convergence results for 1D subdomain coupled DG -> ADCIRC model with minmod limited piecewise linear approximations.

	L_∞ error norms				L_2 error norms			
m	velocity	2^p	elevation	2^p	velocity	2^p	elevation	2^p
1	0.01360372	—	0.01192763	—	0.00477688	—	0.00471115	—
2	0.00195032	6.98	0.00206047	5.79	0.00063685	7.50	0.00057787	8.15
4	0.00006732	29.0	0.00006106	33.7	0.00001305	48.8	0.00001205	48.0
8	0.00002138	3.15	0.00002019	3.02	4.2195e-6	3.09	3.9419e-6	3.06
16	9.3239e-6	2.29	8.6538e-6	2.33	1.8977e-6	2.22	1.7251e-6	2.29
32	4.4057e-6	2.12	4.0281e-6	2.15	9.1884e-7	2.07	8.1627e-7	2.113
64	2.4745e-6	1.78	1.8482e-6	2.18	6.0566e-7	1.52	4.3697e-7	1.87
128	5.6612e-6	0.44	2.5298e-6	0.73	3.0316e-6	0.20	2.0957e-6	0.21
256	0.00001234	0.46	5.6207e-6	0.45	6.8667e-6	0.44	4.8780e-6	0.43
***p* ***		**1.08**		**1.10**		**1.05**		**1.08**

* Converged ratio taken from $m = 32$ ratio before begins to diverge due to ADCIRC BCs.

Table E.16 Temporal convergence results for 1D equation and subdomain coupled DG/ CG -> ADCIRC model with piecewise constant approximations.

	L_∞ error norms				L_2 error norms			
m	velocity	2^p	elevation	2^p	velocity	2^p	elevation	2^p
1	0.00002695	—	0.00001370	—	0.00001420	—	5.9847e-6	—
2	0.00001317	2.05	6.5408e-6	2.09	7.0789e-6	2.01	2.9352e-6	2.04
4	6.5191e-6	2.02	3.1994e-6	2.04	3.5362e-6	2.00	1.4545e-6	2.02
8	3.2483e-6	2.01	1.5814e-6	2.02	1.7702e-6	2.00	7.2461e-7	2.01
16	1.6555e-6	1.96	7.7577e-7	2.04	9.0657e-7	1.95	3.6598e-7	1.98
32	8.5558e-7	1.93	3.7827e-7	2.054	4.7095e-7	1.93	1.8716e-7	1.96
64	7.6128e-7	1.12	2.6870e-7	1.41	4.4192e-7	1.07	2.0573e-7	0.91
128	5.2860e-6	0.14	2.3962e-6	0.11	2.9507e-6	0.15	2.0846e-6	0.10
256	0.00001219	0.43	5.5784e-6	0.43	6.7287e-6	0.44	4.8727e-6	0.43
***p* ***		**0.97**		**1.03**		**0.97**		**0.99**

* Converged ratio taken from $m = 32$ ratio before begins to diverge due to ADCIRC BCs.

Table E.17 Temporal convergence results for 1D equation and subdomain coupled DG/ CG -> ADCIRC model with unlimited piecewise linear approximations.

	L_∞ error norms				L_2 error norms			
m	velocity	2^p	elevation	2^p	velocity	2^p	elevation	2^p
2	1.523e274	—	9.625e273	—	1.056e273	—	1.067e273	—
4	6.861e231	—	1.000e232	—	7.903e230	—	1.158e231	—
8	2.414e212	—	3.329e212	—	2.546e211	—	3.991e211	—
16	2.277e203	—	1.681e203	—	1.781e202	—	1.939e202	—
32	3.168e198	—	4.721e198	—	3.574e197	—	5.475e197	—
64	1.529e196	207.3	2.165e196	218.1	1.861e195	192.0	2.678e195	204.4
128	1.183e195	12.93	1.129e195	19.17	1.167e194	15.95	1.518e194	17.64
256	2.407e194	4.91	2.105e194	5.36	2.164e193	5.39	2.681e193	5.66
512	7.676e193	3.14	6.466e193	3.25	6.667e192	3.25	8.075e192	3.32
1024	3.066e193	2.50	2.538e193	2.55	2.624e192	2.54	3.144e192	2.57
p		**1.32**		**1.35**		**1.35**		**1.36**
**Although unstable, the errors do exhibit a trend and "converge".								

Table E.18 Temporal convergence results for 1D equation and subdomain coupled DG/ CG -> ADCIRC model with minmod limited piecewise linear approximations.

	L_∞ error norms				L_2 error norms			
m	velocity	2^p	elevation	2^p	velocity	2^p	elevation	2^p
1	0.03882913	—	0.02025589	—	0.03191003	—	0.01019256	—
2	0.00174259	22.28	0.000772267	26.237	0.00134152	23.79	0.00037250	27.36
4	0.00004019	43.36	0.00004299	17.97	0.00001048	128.0	0.00001115	33.41
8	0.00001461	2.75	0.00002245	1.91	3.2719e-6	3.20	4.6518e-6	2.40
16	5.9313e-6	2.46	5.9802e-6	3.75	1.1760e-6	2.78	1.2251e-6	3.80
32	2.8130e-6	2.11	2.8294e-6	2.11	5.5308e-7	2.13	5.5781e-7	2.20
64	1.7054e-6	1.65	1.2691e-6	2.23	4.2393e-7	1.30	2.3729e-7	2.35
128	5.1339e-6	0.33	2.4296e-6	0.52	2.9115e-6	0.15	1.9994e-6	0.12
256	0.00001211	0.42	5.5628e-6	0.44	6.7121e-6	0.43	4.8216e-6	0.41
p *		**1.08**		**1.08**		**1.09**		**1.14**
* Converged ratio taken from *m* = 32 ratio, before begins to diverge due to ADCIRC BCs.								